Coreference Resolution via Hypergraph Partitioning

Dissertation
zur
Erlangung der Doktorwürde
der Neuphilologischen Fakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt
von

Jie Cai
aus China

Referent: Prof. Dr. Michael Strube
Korreferent: Prof. Dr. Anette Frank
Einreichung: 08.11 2012
Disputation: 04.04 2013

Abstract

Coreference resolution is one of the most fundamental Natural Language Processing tasks, aiming to identify the coreference relation in texts. The task is to group mentions (i.e. phrases of interest) into sets, so that all mentions in one set refer to the same entity (i.e. a real world object). Mentions are conventionally proper names, common nouns and pronouns. Lately, the coreference task has been extended to deal with verb phrases too. However, we only work with noun phrase mentions in this thesis. By linking mentions together in a document, not only entities are recovered but also different fragments of the context are connected. This therefore leads to a better text understanding. Coreference resolution is essentially important to many applications, such as text summarization and information extraction. In this thesis, we propose a novel coreference model based on hypergraph partitioning. Our system is named ***COPA***, standing for *Coreference Partitioner*. Given a raw document, *COPA* represents it as a hypergraph, upon which the hypergraph partitioning algorithms are applied to derive coreference sets directly.

The Coreference Representation. The coreference relation is a high-dimensional relation, because it depends on multiple types of basic relations (e.g. string similarities and semantic relatedness). Most of the previous work on the coreference resolution task combines the basic relations between mentions into single ones and derives the coreference sets afterward. Since it is relatively expensive to learn the combination of the basic relations, we propose a novel **hypergraph representation model** for coreference resolution. In our model, the mentions are taken as vertices in the hypergraph and the relational features derived from the basic relations as hyperedges. The hypergraph allows for multiple edges between vertices, so that it suits the **high-dimension property** of the coreference relation. Moreover, in a hypergraph one hyperedge can connect more than two vertices. As a result the hypergraph directly represents **the relations between sets of mentions** as required for the coreference resolution task.

Since the basic relations are incorporated in an overlapping manner, *COPA* only needs a few training documents to achieve competitive performance. The **weakly**

supervised nature makes *COPA* a good candidate when applying to different domains or languages, or when only limited training data is available.

The Coreference Inference. The inference of the coreference resolution task deals with sets of mentions. It needs to capture the relations between multiple mentions in order to derive the final coreference sets. Therefore, we consider coreference resolution as a set problem. Most of the previous coreference models address the set problem by dividing the resolution into two steps — a classification step and a clustering step (e.g. Soon et al. (2001)). The classification step makes decisions for each pair of mentions on whether they are coreferent or not. Upon the pairwise decisions, the clustering step further groups mentions into the final sets. The two-step division makes the classification performance not necessarily positively correlated with the end evaluation numbers. It is difficult to track the error propagation and hard to optimize with respect to the final coreference sets. Moreover, since the coreference decisions are made between pairs of mentions independently, global context information is missing in those models.

In this thesis, we propose a global coreference model via **hypergraph partitioning**. We design two algorithms based on the spectral clustering technique — a hierarchical *R2 partitioner* and a flat k-way *flatK partitioner*. We also propose extensions to the clustering algorithms of *COPA*, aiming to include constraints to enforce the cluster-level consistency. The constrained *COPA* is the first attempt towards **a better learning scheme** for our system. It solves the cluster-level inconsistency problem and at the same time contributes to research in the constrained graph clustering field.

The Coreference Evaluation. Since *COPA* is an **end-to-end coreference system**, the important implementation issues encountered when applying clustering algorithms to practical uses are also addressed in this thesis. For instance, the existing evaluation metrics become problematic when the automatically identified mentions do not align with the ones in the ground truth. In this thesis, we propose **variants of the coreference evaluation metrics** to tackle this problem.

COPA outperforms several baseline systems in fair settings, using the same features and the same mentions and only comparing the effectiveness of the models themselves. It also performs competitively compared to the state-of-the-art systems across different evaluation metrics, different data sets and different domains.

Zusammenfassung

Koreferenzresolution ist eine der grundlegendsten Aufgaben der Computerlinguistik. Es wird dabei das Ziel verfolgt, die Koreferenzrelation in Texten zu identifizieren. Die Aufgabe besteht darin, Erwähnungen (d.h. zu untersuchende Phrasen) so in Mengen zu gliedern, dass alle Erwähnungen in einer Menge auf die gleiche Entität (d.h. ein Objekt in der Welt) referieren. Herkömmlicherweise werden Eigennamen, Gattungsnamen und Pronomen zu den Erwähnungen gezählt, wobei in den letzten Jahren auch vermehrt Verbphrasen einbezogen worden sind. In dieser Dissertation werden ausschliesslich nominale und pronominale Erwähnungen berücksichtigt. Indem Erwähnungen in einem Dokument miteinander verknüpft werden, werden nicht nur Entitäten identifiziert, sondern auch verschiedene Kontextfragmente miteinander verbunden. Dies führt zu einem besseren automatischen Textverstehen. Koreferenzresolution ist für viele Anwendungen wie beispielsweise Textzusammenfassung und Informationsextraktion essentiell. In dieser Dissertation schlagen wir ein neues Koreferenzmodell basierend auf Partitionierung von Hypergraphen vor. Unser System heisst ***COPA***, was für *Koreferenz-Partitionierer* (engl. *Coreference Partitioner*) steht. Gegeben ein Textdokument wird dieses in *COPA* als Hypergraph repräsentiert. Anschliessend werden Partitionierungsalgorithmen auf diesen Hypergraphen angewendet, um direkt die Koreferenzmengen zu erhalten.

Die Repräsentation von Koreferenz. Die Koreferenzrelation ist hochdimensional, da sie von vielen Typen von Basisrelationen (z.B. Zeichenkettenähnlichkeiten und semantischer Verwandtschaft) abhängt. Viele frühere Koreferenzresolutionsarbeiten kombinieren verschiedene Basisrelationen zwischen zwei Erwähnungen zu einer einzelnen Relation und treffen die Koreferenzentscheidungen basierend auf diesen kondensierten Relationen. Da es relativ aufwändig ist, die Kombination von Basisrelationen zu lernen, schlagen wir ein neues Repräsentationsmodell basierend auf Hypergraphen für Koreferenzresolution vor. In unserem Modell werden Erwähnungen als Knoten in einem Hypergraphen betrachtet und die Basisrelationen werden als Hyperkanten integriert. Der Hypergraph erlaubt viele Kanten zwischen Knoten, was der hochdimensionalen Eigenschaft der Koreferenzrelation

entspricht. Hinzu kommt, dass in einem Hypergraphen eine Hyperkante mehr als zwei Knoten miteinander verbinden kann. Folglich repräsentiert der Hypergraph direkt die Relationen zwischen Mengen von Erwähnungen, wie es die Koreferenzresolutionsaufgabe erfordert. Da die Basisrelationen überlappend integriert sind, benötigt *COPA* nur wenige Dokumente zum Trainieren, um konkurrenzfähige Ergebnisse zu erzielen. Da *COPA* ein schwach überwachtes Koreferenzsystem ist, eignet es sich auch dann, wenn verschiedene Domänen und Sprachen interessieren oder wenn wenige Trainingsdaten verfügbar sind.

Inferenz für Koreferenz. Die Inferenz für die Koreferenzresolutionsaufgabe erfolgt über Mengen von Erwähnungen. Es müssen dabei die Relationen zwischen mehreren Erwähnungen berücksichtigt werden, um die endgültigen Koreferenzmengen abzuleiten. Wir betrachten daher Koreferenzresolution als ein Mengenproblem. Die meisten bisher vorgeschlagenen Koreferenzmodelle unterteilen das Mengenproblem in zwei Schritte – einen Klassifikationsschritt und einen Clusteringschritt (z.B. Soon et al. (2001)). Im Klassifikationsschritt wird für jedes Paar von Erwähnungen entschieden, ob die entsprechenden Erwähnungen koreferent sind oder nicht. Basierend auf diesen paarweisen Entscheidungen werden die Erwähnungen im Clusteringschritt in die endgültigen Mengen gruppiert. Die Gliederung in zwei Teilschritte führt dazu, dass die Klassifikationsergebnisse nicht notwendigerweise mit den Endresultaten für Koreferenzmengen positiv korreliert sind. Es ist daher schwierig, die Fehlerfortpflanzung zu verstehen und die Inferenz hinsichtlich der endgültigen Koreferenzmengen zu optimieren. Hinzu kommt, dass globale Kontextinformation in diesen Modellen fehlt, da die Koreferenzentscheidungen zwischen Paaren von Erwähnungen unabhängig getroffen werden. In dieser Dissertation schlagen wir ein globales Koreferenzmodell basierend auf Partitionierung von Hypergraphen vor. Wir schlagen zwei Algorithmen vor, die auf der spektralen Clusteringtechnik basieren – ein hierarchischer *R2 Partitionierer* und ein partitionierender *k-way flatk Partitionierer*. Wir präsentieren auch Erweiterungen für die Clusteringalgorithmen von *COPA*, die Nebenbedingungen (engl. *constraints*) einschliessen, um Konsistenz auf der Clusterebene zu erzwingen. Der *constrained COPA* ist ein erster Versuch in Richtung eines besseren Lernschemas für unser System. Es löst spezielle Koreferenzprobleme und trägt gleichzeitig zum Forschungsfeld von Graphclustering mit Nebenbedingungen bei.

Die Evaluation von Koreferenz. Da *COPA* ein Koreferenzsystem mit realen Vorverarbeitungskomponenten ist, befasst sich die vorliegende Dissertation auch mit wichtigen Implementierungsschwierigkeiten, die bei Clusteringalgorithmen auftreten, wenn sie in Anwendungen benutzt werden. So sind beispielsweise Evaluationsmetriken problematisch, da die vom System identifizierten Erwähnungen nicht mit den Erwähnungen im Goldstandard übereinstimmen. Wir schlagen daher in dieser Dissertation neue Varianten der Koreferenzevaluierungsmetriken vor, um mit diesem Problem umgehen zu können.

COPA schlägt verschiedene Baseline-Systeme in einem fairen Evaluierungsszenarium mit gleichen Features, sodass ausschliesslich die Effektivität der Modelle verglichen wird. *COPA* erzielt zudem auch konkurrenzfähige Ergebnisse im Vergleich zu Systemen, welche dem Stand der Forschung entsprechen. Hierbei wird sowohl hinsichtlich verschiedener Evaluationsmetriken als auch in Bezug auf verschiedene Textsammlungen und Domänen verglichen.

Acknowledgments

I always imagined what it takes for me to get all the way through until this point writing my acknowledgment. I learned that it takes four years of learning and working; it takes countless times of producing results worse than the baseline systems; most importantly it takes all the guidance and supports I received during these years.

I am very lucky to have Prof. Dr. Michael Strube as my Ph.D supervisor. The only easy part of this four-year study would have been the communication with Michael, who is always around for giving advices and for listening. I appreciate a lot the freedom he offered and the patience he had even at the time I was very much puzzled and took forever to find my way out. Michael has been a very helpful supervisor and always manages to make complicated things simple for me.

I would also like to thank Prof. Dr. Anette Frank, who took me as her student when I first came to Germany and who now is my Korreferent. I thank Anette for all the warm encouragements and the helpful advices.

The thesis is made much better thanks to the precious efforts my proof-readers have spent. Angela Fahrni literally contributes three pages of German abstract to my thesis; Sebastian Martschat reads so many versions of it and helps to improve both the contents and the writing a lot; Yufang Hou has been always carefully reading and provides me with many constructive suggestions. Besides my Ph.D sisters and brothers, Dr. Camille Guinaudeau gives me a lot of helpful review comments; my English proof-reader Dr. Jiawei Mao corrects thoroughly through the entire thesis; my friend Shiyang Lu sends me the detailed reviews from far-away the other half of the earth. Since I know by heart how busy all my kind proof-readers are, I appreciate so much the time each of them squeezed out for my thesis.

My family and friends have been constantly encouraging, supporting and inspiring me throughout the four years. My parents Houming Cai and Jieyun Liu are there for me as they always are, and they make things so easy by being such cheerful parents. I thank my husband Yangyang Zhao for all the phone calls which accompanied me walking down the hill in the dark forest, for all the good memories we

had in Europe and for being very supportive even when I got very cranky during the Ph.D study. I here thank Angela again, who has been at my side (literally again) for talking through the confusions, for discussing about researches and for the casually gossipings. I also would like to thank my dear friends Lingling Kong and Zhen Zeng for simply everything.

There are so much more I should mention and so many more people who helped. HITS gGmbH supports my Ph.D program; Heidelberg city gives me a German home; things would have been much harder for me at the beginning if it were not for the help of my former colleague Viola Ganter; I still remembered my first amazing trip thanks to my favorite postdoc Dr. Vivi Nastase.

At the end, I want to thank myself too. Cai, I am happy to see that you learned a lot along the way. I appreciate the hard work you did for these years and especially that you managed to keep jogging at the same time. I am glad that you had a good time.

Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst und keine anderen als die ausdrücklich genannten Quellen und Hilfsmittel verwendet zu haben.

(Jie Cai)

Contents

Chapter 1

Introduction

”Hi Cai,
you must be very brave to work on
Coreference Resolution.”
– Prof. Mirella Lapata[1] –

This thesis addresses the challenge of within-document **coreference resolution**, a task of grouping the referring expressions (i.e. phrases) of entities (i.e. real world semantic objects) into coreference sets so that all expressions in one set refer to the same entity. The coreference relation is dependent on multiple basic relations such as the shallow syntactic relation and semantic relatedness. It can be derived from one of the basic relations or from a combination of multiple ones, depending on different contexts. Therefore we consider the coreference relation as **a complex relation and a high-dimensional relation**, as opposed to the basic low-dimensional relations. Since the coreference resolution task is not only to detect the pairwise coreference relation but also to group the referring expressions into sets, we consider the task as **a set problem**. By analyzing the linguistic phenomena of the coreference relation and understanding the task requirements, we raise four important questions which are addressed throughout the thesis — (1) representing the coreference relation, (2) inferring the coreference relation, (3) evaluating coreference resolution, (4) learning cheaply.

Our proposed coreference model is motivated by the first two questions. Both its representation model and its inference method address the requirements (1) and (2) correspondingly. Our model represents documents as **hypergraphs**, which allow for multiple edges between vertices and multiple vertices within one edge. The vertices are the referring expressions from the documents, and the multiple edges between them enable us to break down the complex coreference relation into multiple basic ones. Moreover, the hyperedges containing

multiple vertices straightforwardly represent the sets of expressions. Upon the hypergraph representation, we apply **graph partitioning** techniques to partition the hypergraphs into sub-hypergraphs, each of which corresponds to a coreference set. Our system is named ***COPA***, standing for Coreference Partitioner. *COPA* differs significantly from the previous local models, since it is able to take the global context (of a document) into consideration and to generate the coreference sets simultaneously in one step.

We work on **an end-to-end system setting**, which takes raw texts as input and extracts coreference sets in a fully automatic way. Since the presence of noise is unavoidable in such a realistic setup, not only the modeling itself but also the practical issues are addressed in this thesis. For instance, our proposed evaluation metrics aim to conquer the problems of the widely used metrics when evaluating the noisy output from end-to-end coreference systems.

In this chapter, we start with introducing the coreference phenomena from a linguistic point of view in Section 1.1. Section 1.2 then describes the coreference resolution task and the four questions consequently emerging. In Section 1.3, we convey the intuitions behind our proposal of *COPA* and the main contributions of the thesis. The general structure of the thesis is given at the end in Section 1.4.

1.1 Anaphora and Coreference

In linguistic expressions, in order to preserve the coherence in texts while keeping the diverse phrasal expressions at the same time, the referring expressions are used frequently. In the following Example (1), the pronouns [*him*], [*he*] and [*his*] are all referring expressions, which are called **anaphors** or **anaphoric expressions**. An anaphor is used to refer to an **antecedent** which is a preceding phrase (e.g. [*Yemen's President*]), and they are talking about the same object in the world. A world object is called an **entity**, for instance the YEMEN'S PRESIDENT in Example (1)[2]. The process of identifying the correct antecedent for an anaphor is **anaphora resolution**.

> Example (1): [*Yemen's President*]$_1$ has repeatedly said an internal explosion rocked the "USS Cole", but tomorrow the U.S. official expects [*him*]$_1$ to announce that [*he*]$_1$ has changed [*his*]$_1$ mind, and tomorrow, the search for bodies will resume .

Besides the pronominal anaphors, as shown in Example (1), definite and demonstrative phrases are often used as the anaphoric expressions too (e.g. [*the meeting*] and [*the regulators*] in Example (2)). Proper names can either mention a new entity or refer to a previous one, such as both mentions of [*Lincoln*].

[2]The entities are in capitalized fonts throughout this thesis.

Example (2): In [*a highly unusual meeting in Sen. DeConcini 's office in April 1987*]$_1$, the five senators asked [*federal regulators*]$_2$ to ease up on [*Lincoln*]$_3$.

According to notes taken by one of the participants at [*the meeting*]$_1$, [*the regulators*]$_2$ said [*Lincoln*]$_3$ was gambling dangerously with depositors ' federally insured money and was " a ticking time bomb ."

An anaphor and its antecedent are said to be **coreferent** with each other. In other words, both of them are linguistic expressions that refer to a specific entity. It is common that there are multiple linguistic expressions for an entity in a document, which together form a **coreference chain** or a **coreference set** (e.g. all the phrases marked with the same subscripts in Example (1) form one coreference set). The process of identifying the coreference sets within or across documents is **coreference resolution**. As Example (2) illustrates, a document tends to have multiple coreference sets, and coreference resolution is to identify all of them commonly.

Coreference resolution is closely related to anaphora resolution, and it can be viewed as a post-processing upon the antecedent-anaphor output from anaphora resolution. Considering Example (1), resolving [*him*], [*he*] and [*his*] to [*Yemen's President*] respectively during anaphora resolution will help to generate the entire coreference set. However, in this thesis, we argue that global (set-level) information is missed from such post-processing interpretation. In the same Example (1), when the first two pronouns are resolved to the entity YEMEN'S PRESIDENT, it is more likely for the third one to refer to this salient entity too rather than to the entity THE U.S. OFFICIAL. As a result, a set-based one-step coreference resolution model is preferable due to its global property.

1.2 The Coreference Resolution Task

In this section, the crucial requirements for modeling the coreference resolution task are discussed within an end-to-end system framework. Our proposed coreference model is motivated by the requirements and addresses all of them throughout the thesis.

The coreference resolution task is to group the referring expressions into sets so that all expressions in one set refer to the same entity. An end-to-end coreference system takes raw documents as input and generates the identified coreference sets as output, via a pipeline of automatic processors. Figure 1.1 shows an example text displayed in *MMAX*, which is a multi-layer visualization tool to help illustrate the coreference examples (Müller & Strube, 2006).

The phrases that need to be resolved for coreference resolution are conventionally called **mentions** in the task, such as [*Gore*], [*I*], [*he*], [*his opponent*] and [*the vice president*]. In this thesis, the mentions marked with square brackets (i.e. []) are **true mentions**, which are taken from the ground truth annotation, and the ones in curly brackets (i.e. {}) are **system mentions**, which are derived automatically. The running **entity** in this example is GORE,

whose corresponding coreference set is {[*Gore*], [*I*], [*he*], [*his opponent*], [*the vice president*], ...}.

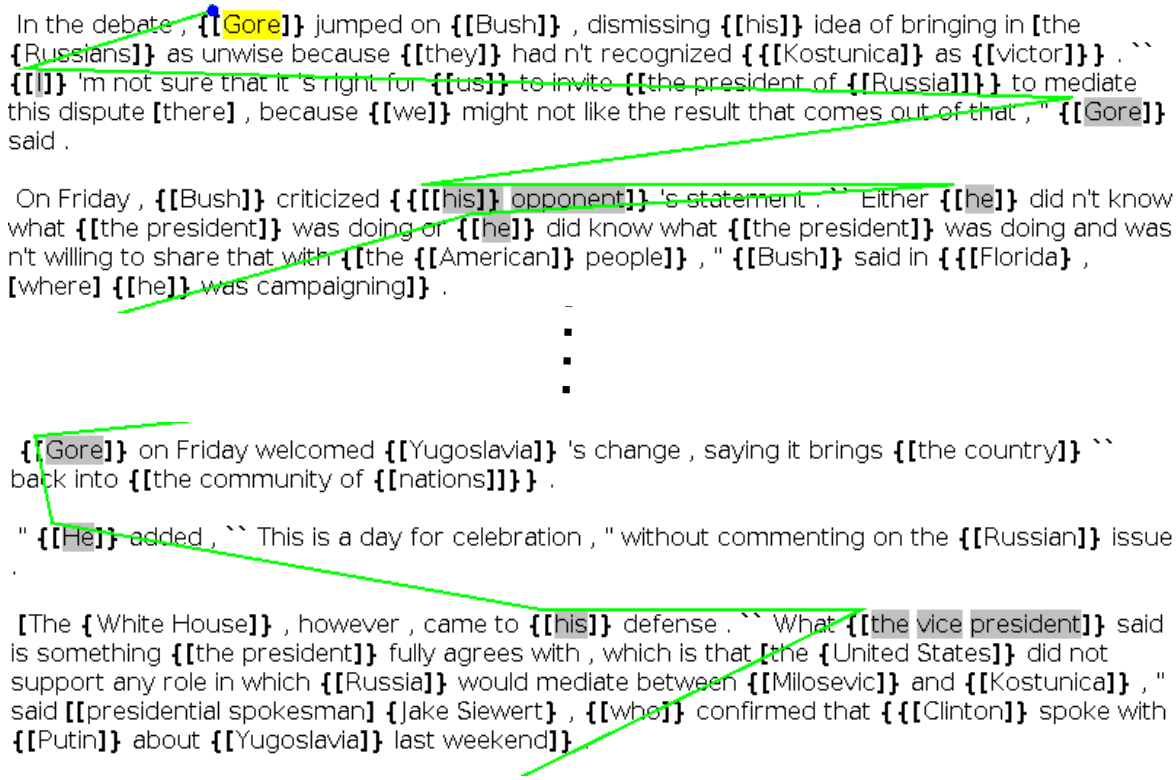

Figure 1.1: Example (3): Coreference Resolution in *MMAX*

The pre-processing components may vary between different systems, but the most important ones are sentence splitting, POS tagging, mention detection and syntactic parsing. The pre-processors provide a coreference system with the mentions to be resolved and contextual information for assisting the resolution procedure. When external resources are available, more components for knowledge extraction may be incorporated into the system accordingly.

The following subsections will introduce the most important aspects for designing a coreference system.

1.2.1 Representing the Coreference Relation

The Coreference relation is a high-dimensional relation. By interpreting the coreference relation as a high-dimensional relation, we refer to the fact that the coreference relation is dependent on different types of basic relations, such as shallow syntactic dependency and semantic relatedness. These basic relations are considered to be low-dimensional, which together form the (more) complex coreference relation.

We use the same Example (3) (Figure 1.2) in this subsection to convey the *high-dimension* property of the coreference relation. It can be seen that within the exemplar text, there are several diverse basic (low-dimensional) relations which comprise the coreference relation.

In Example (3), with the entity GORE, the coreference relation between the first [*Gore*] and the second [*Gore*] can be easily detected just based on their high string similarity. However, in order to resolve the coreference relation between [*Gore*] and [*the vice president*], external knowledge resources are necessary for providing relevant information about vice president GORE. If it has been mentioned in the preceeding text that GORE is a vice president (e.g. in a text fragment "the Vice President Gore"), the relation can be also retrieved from the very text by extracting the relevant attributes for the entity GORE before the resolution.

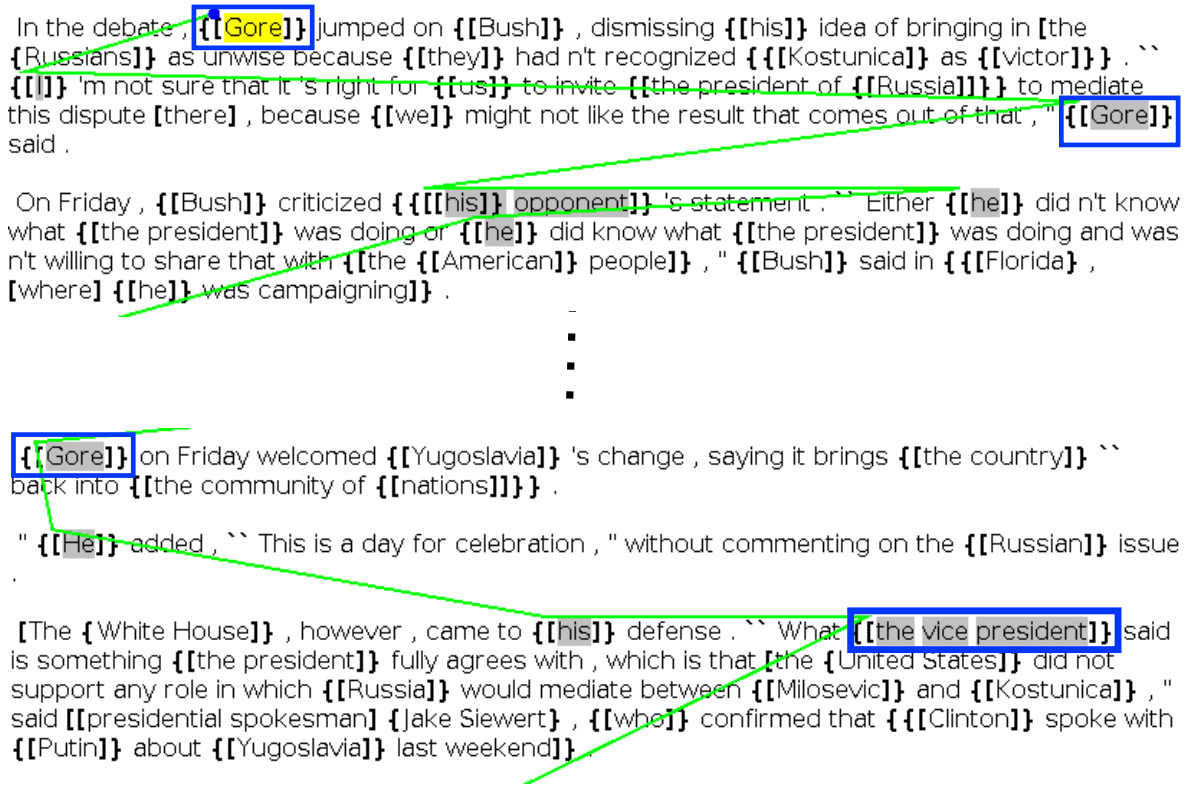

Figure 1.2: Example (3): Coreference Relation is High-Dimensional (part 1)

For the same Example (3), Figure 1.3 illustrates a more complex coreference relation between the mentions [*his opponent*] and [*Gore*], whose resolution requires a reasoning scheme upon the two entities GORE and BUSH. In order to identify the relation between [*his opponent*] and [*Gore*] correctly, it is necessary to resolve [*his*] to [*Bush*] at first and afterward to extract the fact that GORE is the opponent of BUSH in the debate. In this case, the coreference relation is much more complex than the ones between mentions which share the same strings.

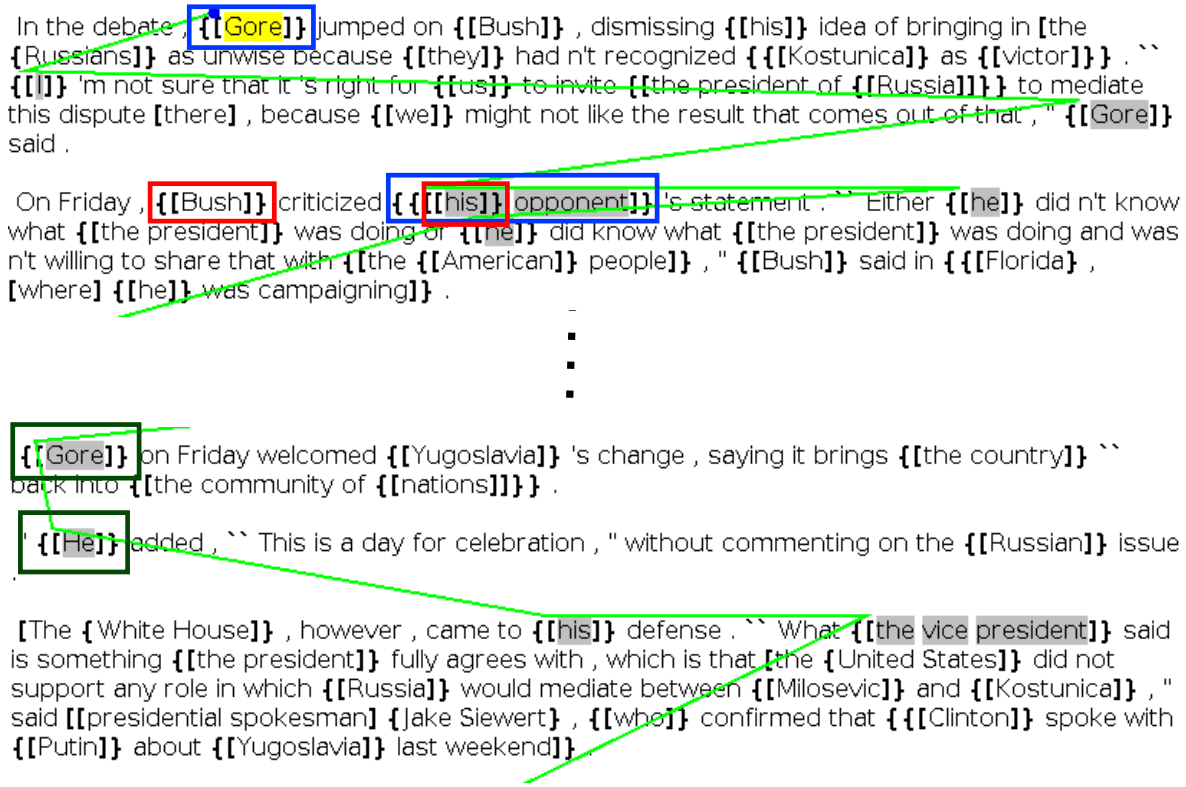

Figure 1.3: Example (3): Coreference Relation is High-Dimensional (part 2)

The coreference relation of pronouns is often based on local phenomena. Considering the pronoun [*He*] marked in Figure 1.3, which is in a parallel sentential structure with [*Gore*], i.e. "[*Gore*] said" and "[*He*] added". It is reasonably confident for such a structural relation to indicate the coreference relation for pronouns. However, structural information is a much weaker indicator for most of the non-pronominal anaphors.

To sum up, the coreference relation can be inferred from multiple low-dimensional relations (e.g. string match and parallel structure). Depending on the types of the participating mentions and the local contexts, different basic relations can be dominating or be interacting with each other during coreference resolution.

Q1: How to represent the multiple low-dimensional relations and to allow their interactions?

is the first question to consider in terms of the representation model for a coreference resolution system.

1.2.2 Inferring the Coreference Relation

The coreference resolution task is a set problem. The coreference resolution task is to group mentions into disjoint coreference sets, so that each set corresponds to an entity. The resolution decision for one mention depends on the resolutions of all the others in the same text, which together provide the global context for the mention in focus. As explained for Example (3) (Figure 1.3), the resolution of the mention [*his opponent*] should benefit from the resolution of the embedded mention [*his*]. Therefore, inferring the coreference sets simultaneously is essential to making use of the complete context.

In order to achieve the overall optimized coreference sets, the inference procedure needs to consider not only the relations between mentions within the same sets, but also the relations between mentions from different sets. Since the optimization is conducted at the output end, it is important to preserve all relations from a document until the final generation of the coreference sets. Hence it is preferred to have the coreference sets identified directly from the original relations.

Q2: How to derive coreference sets directly and simultaneously?

is the second crucial question we need to consider. It regards the choice of the inference algorithm.

1.2.3 Evaluating Coreference Resolution

Evaluating the system output sets against the true coreference sets is no trivial matter. There have been several evaluation metrics designed for the coreference resolution task, either evaluating on mention pairs or on sets directly. However, they become problematic in a realistic system setup, where the system mentions do not align with the true mentions any more.

Q3: How to evaluate end-to-end coreference resolution systems?

is the third concern of ours in this thesis.

1.2.4 Cheap Learning?

There are several data sets proposed for evaluating coreference resolution systems, most of which are collections of news articles, such as the examples illustrated in this section. Since

the coreference relation is a general linguistic phenomenon, coreference resolution is applicable to different domains (e.g. the medical domain) and to different languages. This urges the requirements of a large amount of annotated data sets for the purpose of the model training. Annotating the corpora manually is considered to be expensive, therefore the question

Q4: Can we use less training data?

becomes important when extending the coreference system to open domain texts or when applying the system to multilingual tasks.

1.3 Contributions of this Thesis

Most recent approaches to coreference resolution divide the task into two steps: (1) a classification step which determines whether a pair of mentions is coreferent or which outputs a confidence value for this pair, and (2) a clustering step which groups mentions into entities based on the output of step 1.

In this thesis, we propose a global one-step model — *COPA* — to approach the coreference resolution task. *COPA* is a novel coreference model which avoids the division into two steps and instead performs a global decision in one step. It represents a document as a hypergraph, where the vertices denote the mentions and the edges denote the (low-dimensional) relational features between mentions. Coreference resolution is performed globally in one step by partitioning the hypergraph into sub-hypergraphs so that all mentions in one sub-hypergraph refer to the same entity. The left part of Figure 1.4 illustrates the appearance of the hypergraph built by *COPA* and the right part shows the *COPA* output after the partitioning procedure. This example is described in more detail in Chapter 4.

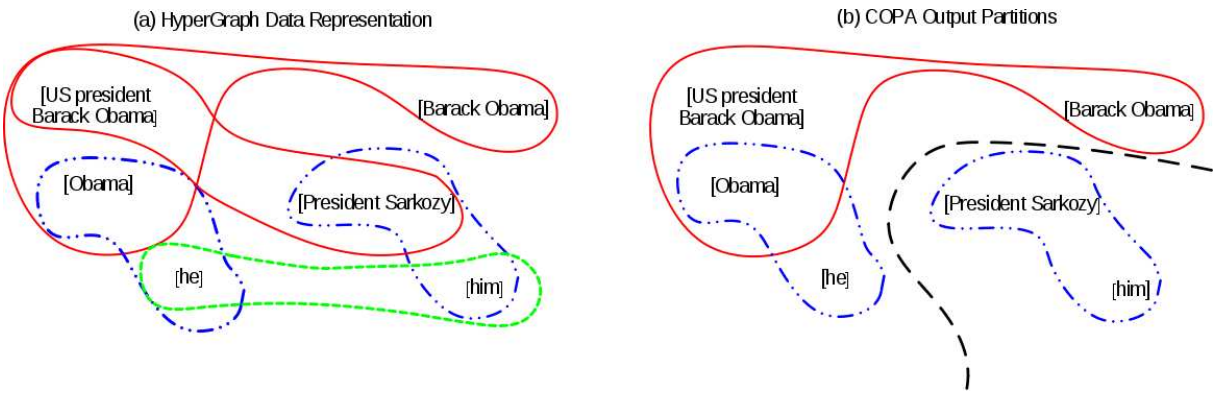

Figure 1.4: COPA Example: Processing Illustration

With *COPA*, we are able to address the four questions raised in Section 1.2, which are explicated in Section 1.3.1 to Section 1.3.4.

1.3.1 Representing the Coreference Relation

Previous two-step models attempt to predict a single confidence value between a pair of mentions by learning the combination of features from the training data (Soon et al., 2001; Luo et al., 2004; Rahman & Ng, 2009; Bengtson & Roth, 2008). Since these models base their clustering step on the collapsed relations, some global information which could have guided step 2 is already lost. In the other hand, global information cannot be accessed in step 1 when making the pairwise decisions.

The hypergraph representation of *COPA* (e.g. Figure 1.4 (a)) enables the multiple relational features to directly come in (as hyperedges) without the necessity of collapsing them into single ones (as standard edges) as standard graph models would have to. Comparing with the standard graph, the hypergraph has additional representation power. A hyperedge connects two or more than two vertices (e.g. the hyperedge connecting [*Obama*], [*US president Barack Obama*] and [*Barack Obama*]), and between vertices there can be multiple hyperedges involved (for the sake of a clear illustration, Figure 1.4 does not include overlapping hyperedges). The set property and the overlapping manner of hyperedges make the hypergraph a good candidate for representing the coreference relation. In brief, the hypergraph allows for **representing multiple low-dimensional relations and capturing set-level information**, so that the representation model of *COPA* is **intuitively representing coreference phenomena**.

Moreover, since the hypergraph is a generalization of the standard graph, the algorithms based on standard graphs are still applicable to hypergraphs with necessary adaptations. It is easy to include more relations as hyperedges into the hypergraph model and various graph-based inference algorithms are supported on top of the *COPA* model.

1.3.2 Inferring the Coreference Relation

For most of the two-step methods, the classification steps vary in the choices of the classifiers and the numbers of features used. The clustering step exhibits much more variations: Local variants utilize greedy search strategies (Soon et al., 2001; Ng & Cardie, 2002) while global variants optimize globally but still upon the pairwise output from step 1 (Luo et al., 2004; Daumé III & Marcu, 2005; Nicolae & Nicolae, 2006; Denis & Baldridge, 2009). As already mentioned, since these methods base their global clustering step on a local pairwise model, some global information which could have guided step 2 is already lost.

There have also been attempts on establishing global one-step models, most of which are probabilistic ones (Culotta et al., 2007; Sapena et al., 2010; McCallum & Wellner, 2005;

Poon & Domingos, 2008). The global models allow one to make use of set-level information and more context during the inference procedure.

Upon the hypergraph representation, *COPA* applies graph partitioning techniques to derive coreference sets directly and simultaneously. The graph partitioning algorithms of *COPA* generate the optimized coreference sets, so that the mentions within the same sets are connected to each other as closely as possible, while the mentions from different sets as loosely as possible. It is the first graph-partitioning-based coreference model that takes all mentions from a document into one unified graph and achieves competitive performances across different data sets in a realistic setting. Partitioning algorithms enable us to make a **global coreference decision** by using whichever contextual information encoded in the graph, rather than to work in a sequential and local manner.

Unlike the probabilistic models, *COPA* is based on a graph partitioning technique that is preferable for its simple inference procedure. We differ from Nicolae & Nicolae's graph partitioning model (Nicolae & Nicolae, 2006), as we do not make pairwise coreference predictions and we manage to handle all types of mentions in one unified model.

1.3.3 Evaluating Coreference Resolution

In this thesis, we address an important issue in the coreference resolution task — evaluation metrics. Since most widely used metrics are designed to handle true mentions only, they become problematic when evaluating end-to-end coreference systems. We propose **variants of different evaluation metrics** for dealing with this issue.

1.3.4 Cheap Learning!

The hypergraph-based coreference model of *COPA* derives the coreference relation by analyzing the graph structure at the inference phase, and the relational features used for the graph construction are simply represented in an overlapping manner. Since no feature combination function needs to be learned beforehand, *COPA* only requires a small amount of training data to learn the weights for low-dimensional relations (i.e. hyperedge weights), which makes *COPA* a **weakly supervised** system.

1.3.5 Other Contributions

Coreference resolution is a set problem and thus the coreference relation is a transitive relation. Due to the transitive closure which is implicitly done during the partitioning process of *COPA*, inconsistent coreference sets may be derived. Different optimization strategies have been employed in the literature in order to enforce the coreference transitivity. In this thesis,

we address this problem within the graph partitioning framework by proposing constrained clustering algorithms. We propose a novel method to combine constrained data clustering algorithms with the spectral graph clustering technique via the spectral embedding, hereby contributing to the constrained graph clustering field. At the same time, the **constrained *COPA*** contributes to the coreference problems which can only be solved by considering cluster-level consistencies. We experiment with both artificial clean constraints and automatically generated ones. Although the clean setting produces promising improvements, our results on the automatically generated constraints are mostly negative for now. Further efforts on designing more high-recall constraints are needed.

Extensive experiments show that *COPA* outperforms strong baseline systems in strict fair comparisons, and it performs competitively with a small feature set and a small amount of training data across different domains.

1.4 The Thesis Structure

The thesis is organized into two parts, (1) Chapter 1 to Chapter 7 form the backbone of our contributions to the coreference resolution task; (2) Chapter 8 introduces the important extensions we made upon the basic version of *COPA* model, both in the algorithms and in solving special types of coreference problems.

- Chapter 1 helps the readers to develop an idea about the work presented in this thesis — the motivation and the significant contributions.
- Chapter 2 introduces the important related work for coreference resolution, which provides a big picture to the task modeling.
- Chapter 3 describes the corpora used throughout the thesis. The annotation schemes adopted by each of the data sets are illustrated and the important differences between them are pointed out. The chapter aims at assisting the reader to get familiar with the coreference phenomena and the involved issues related to annotations, both of which are important for understanding the coreference resolution task addressed in this thesis.
- Chapter 4 introduces our proposed coreference system — *COPA*. The chapter is self-contained, with the representation model, the partitioning algorithms and the system components described in detail. For the techniques involved in the basic version of *COPA*, readers can read Chapter 4 and Chapter 7 (for experiments) alone, with the features in Chapter 5 to be briefly looked up if necessary.
- Chapter 5 presents the features used in *COPA*.

- Chapter 6 discusses the problems of the previous evaluation metrics and then introduces our variants of the metrics for evaluating end-to-end coreference systems. Experiments verifying our variants are included at the end of the chapter. For readers who have been working in the field and are concerned about evaluating end-to-end coreference systems due to ones' own experiences, Chapter 6 can be read as a stand-alone chapter.

- Chapter 7 evaluates *COPA* with thorough experimental comparisons, against strong baseline systems and state-of-the-art systems in different domains.

- Chapter 8 describes the constrained version of *COPA*. We aim to guide the system towards more consistent partitionings by imposing negative (i.e. *Cannot-Link*) constraints on the partitioning algorithm. Experimental results for constrained *COPA* are provided within the chapter.

  For readers interested in graph clustering algorithms, Chapter 8 focuses on including constraints into graph clustering algorithms without changing the objective functions, and the chapter applies the proposed methods to an application of coreference resolution. Readers may also want to check on all the implementation issues addressed in Chapter 4, which give important hints to use clustering techniques for real applications.

- Chapter 9 concludes the entire thesis and suggests future improvement directions for graph-based coreference models.

1.5 Published Work

The proposal of *COPA* is published in (Cai & Strube, 2010a), where the hypergraph representation of texts and the coreference inference via partitioning are described. (Cai et al., 2011b) describes the positive-negative-weak feature engineering and illustrates the application of *COPA* on a large corpora to compete with the state-of-the-art systems. *COPA*'s participation on clinical tasks is introduced in (Cai et al., 2011a).

The proposed evaluation metrics for end-to-end coreference resolution are published in (Cai & Strube, 2010b).

Chapter 2

Related Work On Coreference Models

Understanding and automatically resolving the coreference phenomena in texts has been of interest to computational linguists for decades, starting from the early work on linguistic theories to the latest research on exploring machine learning techniques. The inclusion of the early theories (Section 2.1) in this chapter is to illustrate the linguistic insights they provide, which still inspire good features for modern methods. However, the main stream of research is moving towards the machine-learning-based task modeling (Section 2.3 to 2.5).

In this chapter, the most important research lines in the field are introduced. The existing coreference models are categorized according to their learning schemes — rule-based systems (Section 2.2), unsupervised models (Section 2.3), weakly supervised methods (Section 2.4) and finally the supervised ones (Section 2.5). Our proposed system is **a supervised coreference model**. However, we show in Chapter 7 that our system only needs a little training data to achieve competitive performance, which makes it a weakly supervised one (when using limited training data). Unlike the weakly supervised methods in Section 2.4 which make use of unlabeled data together with labeled ones, our model is only trained on (manually) annotated data in a conventional supervised manner without making bootstrapping procedures necessary.

2.1 Early Theories and Formalisms

In this section, two important theories related to coreference resolution are introduced. Centering theory (Section 2.1.1) studies the referring relation between utterances (e.g. sentences) and entities in order to model the discourse coherence. This theory can be used directly to estimate the possible entity assignments for referring expressions, and therefore to predict the coreference relation. Centering theory is summarized with its important claims in this section, and the details are provided in the corresponding references.

Binding theory (Section 2.1.2) models the preference of antecedents for anaphoric expres-

sions on dependency trees. It can be easily adopted as relational features (or constraints) for machine-learning-based coreference models.

2.1.1 Centering

Centering theory (Grosz & Sidner, 1986; Grosz et al., 1995; Strube & Hahn, 1999) is a theory of the local component of attentional state. Joshi & Kuhn (1979), Joshi & Weinstein (1981) show that there is a connection between changes in immediate focus and the complexity of the inference required for understanding the utterances in the corresponding discourse. From a coreference modeling point of view, the less complex the required inference is, the more possible it is to be a correct usage of referring expressions in the utterances.

Centers (e.g. referring expressions) of an utterance refer to the entities which help to link the utterance to others within a discourse segment. Each utterance U in a discourse segment DS has a set of *forward-looking centers*, $C_f(U, DS)$ and (except for the segment initial utterance) has a single *backward-looking center*, $C_b(U, DS)$. The simplified notations are $C_f(U)$ and $C_b(U)$. When a center c is the semantic interpretation of an utterance U, it is defined as a relation — U *directly realizes* c. A "*realizes*" relation is a generalization of the "*directly realizes*". Since the realization relation combines syntactic, semantic, discourse, and intentional factors, the centers of an utterance are determined by the properties of the utterance in focus, the corresponding discourse segment and the discourse.

The center elements of $C_f(U_n)$ are derived from the expressions that constitute U_n, and they are partially ordered according to their prominences in U_n. The top ranked element of $C_f(U_n)$ that is also realized in U_{n+1} is taken as $C_b(U_{n+1})$. Three types of transition relations between pairs of utterances are defined.

1. *Center continuation:* $C_b(U_{n+1}) = C_b(U_n)$, and the entity is the top ranked element of $C_f(U_{n+1})$.

2. *Center retaining:* $C_b(U_{n+1}) = C_b(U_n)$, but this entity is not the top ranked element in $C_f(U_{n+1})$.

3. *Center shifting:* $C_b(U_{n+1}) \neq C_b(U_n)$.

Different centering transitions between utterances indicate different degrees in coherence for the corresponding segment. The most fundamental claim of centering theory is that the inference load on the hearer decreases as the discourse coherence increases. Several other major claims are provided, which can be used as constraints for coreference modeling.

1. *A unique* C_b*:* each U_n has only one backward-looking center.

2. *Ranking of C_f:* the elements of C_f are partially ordered according to a number of factors.

3. *Centering constraints realization possibilities:* if any element of $C_f(U_n)$ is realized by a pronoun in U_{n+1}, then $C_b(U_{n+1})$ must be realized by the pronoun too.

4. *Preferences among sequences of center transitions:* sequences of continuation are preferred over sequences of retaining; sequences of retaining are to be preferred over sequences of shifting.

5. *Primacy of partial information:* a semantic theory supporting the construction of partial interpretations is necessary.

6. *Locality of $C_b(U_n)$:* $C_b(U_n)$ cannot be corresponding to $C_f(U_{n-2})$ or other prior sets of forward-looking centers.

7. *Centering is controlled by a combination of discourse factors:* centers are determined on the basis of the combination of syntactic, semantic and pragmatic processes.

Centering theory connects the focuses of attention, the choices of referring expressions, and the coherence of utterances within a discourse segment. It has been used in extended or re-formulated forms for anaphora resolution tasks (Brennan et al., 1987; Hahn & Strube, 1997; Strube, 1998; Walker, 1998).

2.1.2 Binding Theory

The binding theory is formulated in Chomsky's Lectures of Government Binding (Chomsky, 1981; Chomsky, 1995), which discusses anaphora within the generative paradigm. It considers the anaphoric relation for reflexive pronouns, reciprocals, personal pronouns and referential expressions (lexical noun phrases), by imposing syntactic constraints on their NP interpretations. Reflexives and reciprocals need local antecedents; pronouns may have an antecedent, but must be free locally; referential expressions must be free.

The three principles in binding theory are described as:

Principle A: An anaphor (reflexive or reciprocal) must be bound in its governing category.

Example: *[John]*$_i$ saw *[himself]*$_i$. (*[John]* binds *[himself]*, and they are coreferential.)

Principle B: A pronoun (except reflexive and reciprocal) must be free in its governing category.

Example: *[John]*$_i$ saw *[him]*$_j$. (*[John]* binds *[him]* which violates the principle, so that they are not coreferential.)

Principle C: An referential expressions must be free everywhere.

Example: *[John]*$_i$ saw *[Katja]*$_j$. (*[John]* binds *[Katja]* which violates the principle, so that they are not coreferential.)

The binding theory is helpful in ruling out the antecedents for pronominal anaphors that violate the proposed constraints, as well as in assigning possible antecedents to bound anaphors. For instance, our feature (6) corresponds to Principle C and feature (17) to Principle A (see Chapter 5).

2.2 Rule-based Deterministic Coreference Models

The coreference resolution systems from earlier years (e.g. Hobbs (1978) and Lappin & Leass (1994)) rely on manually configured rules, most of which are derived from the linguistic interpretations of the coreference phenomena. There are a couple of lately emerged coreference resolution systems (Section 2.2.3 and 2.2.4) which are also completely built upon heuristic rules and perform in a deterministic manner. These systems aim to explore how syntactic and semantic information helps the task by neglecting the effect of the learning schemes. The successfully explored heuristic rules should inspire (strong) features for machine-learning-based algorithms (see Section 2.3, 2.4 and 2.5), and the deterministic systems may serve as good baselines for the complex coreference models.

2.2.1 Hobbs' Algorithm

Hobbs (1978) proposes one of the first algorithmic approaches to pronoun resolution, determining the antecedents for pronominal anaphors by searching on syntactic parse trees and incorporating semantic analysis.

Hobbs' first algorithm performs on surface parse trees, which are assumed to be correctly available for each sentence to be resolved. A surface parse tree exhibits the grammatical structure of a sentence. This simple method traverses the tree in a particular order looking for a noun phrase of the correct gender and number as the expected antecedent of a pronoun. Selectional constraints can be further applied to the algorithm to restrict the candidate antecedents.

Hobbs' second algorithm is working on texts, where the syntactically derivable coreference and non-coreference relations have already been detected. The texts should be in logical representations, exhibiting functional semantic relationships. In this semantic algorithm, there are four principal semantic operations on logical notations of texts. These are (1) detecting

inter-sentence connections, (2) interpreting general words or predicates in context, (3) merging redundant statements and (4) extracting the yet unidentified entities. The four options together are able to accomplish the pronoun resolution most of the times. Where they fail, the naive algorithm is used to determine the final antecedent.

Hobbs' approach remains one of the most influential work in the field and serves frequently as a common benchmark for evaluating later proposals (Mitkov, 2002).

2.2.2 Lappin and Leass' Algorithm

Lappin & Leass (1994) propose an algorithm, *RAP* (Resolution of Anaphora Procedure), which is applied to the syntactic representations generated by McCord's Slot Grammer parser (McCord, 1989). The system uses multiple salience measures, which capture a variety of syntactic properties. Moreover, the system uses a model of attentional state too.

From a list of candidate antecedents of a pronominal anaphor, *RAP* determines the preferred one by relying on several components.

1. An intra-sentential syntactic filter.
2. A morphological filter rules out the candidate NPs for a pronoun according to their syntactic grounds or agreements on person, number or gender.
3. Pleonastic pronouns are identified by a separated filter.
4. An NP is assigned several salience values, which favor (i) subject over non-subject NPs, (ii) direct objects over other complements, (iii) arguments of a verb over adjuncts and objects of prepositional phrase adjuncts of the verb, and (iv) head nouns over complements of head nouns.
5. For an equivalent class of NPs, an overall salience value is calculated.
6. At the end, a decision maker selects the preferred antecedents for each anaphoric pronouns

Lappin & Leass test *RAP* on five computer manuals containing approximately 82,000 tokens. The success rate of the system is optimized on the training set in a heuristic way. In the blind test, *RAP* scores higher than a Slot Grammer version of Hobbs' algorithm (Hobbs, 1978).

2.2.3 Haghighi and Klein's Simple System

Haghighi & Klein (2009) present a deterministic coreference system, which is driven by syntactic and semantic compatibility lists extracted from an unlabeled corpus. They try to break from the standard view of focusing on coreference modeling. Instead, they are devoted to exploring **linguistic features** in a simple deterministic manner.

Haghighi and Klein's system works in a three-step process. For each anaphor, a best antecedent is chosen or is set to be NULL, following the three steps:

1. **Syntactic Constraints**: a self-contained syntactic module generates syntactic structures using an augmented parser and extracts syntactic paths from the anaphor to its candidate antecedents. When applicable, syntactic constraints either enforce or disallow coreference relations on paths.

2. **Semantic Constraints**: a self-contained semantic module evaluates semantic compatibilities between head words and between names, so that this module further filters the remaining antecedents from 1.

3. **Selection**: Select the final antecedent with the minimal tree distance to the anaphor.

For agreement constraints, Haghighi & Klein implement *person*, *number* and *entity type agreements*. *Role appositives* and *predicate nominatives* are extracted from syntactic trees to assist non-pronominal resolution. A set of compatible word pairs which match the predicate-nominative patterns are extracted from two external data sets, so that rich semantic knowledge can be accessed.

The simple system manages to outperform the state-of-the-art unsupervised coreference resolution systems and is broadly comparable to the state-of-the-art supervised systems. The authors suggest to use the system as a simple-to-reproduce and high-performance baseline for future work in the field.

2.2.4 Stanford's Multi-Pass Sieve System

When participating in the CoNLL-2011 shared task (Pradhan et al., 2011) which is one of the most influential shared task on the coreference resolution task, the Stanford's system (Lee et al., 2011) won in all provided settings. The proposed *Multi-pass Sieve* system is built in an architecture which implements multiple sieves in a cascaded manner. In a top down manner, the sieves output the highest precision predictions to the lowest ones. Since at each sieve all information available (including the predictions from previous sieves) can be used, **cluster-level features** (e.g. *cluster head match*) have a means to come into the model. The sieves proposed are described briefly below.

1. **Pass 1:** Extract string match.
2. **Pass 2:** Precise constructions (e.g. appositive; predicate nominative; role appositive).
3. **Pass 3:** Strict Head Match (e.g. cluster head match; compatible modifiers).
4. **Pass 4 & 5 & 6:** Variants of head match.
5. **Pass 7:** Pronoun resolution.

Despite of its simplicity, Stanford's multi-sieve system achieves more competitive performance than most of the complex models. With careful engineering, it is easier to add more sieves and features without harming the performance which on the other hand can frequently happen to more sophisticated models.

2.3 Unsupervised Coreference Models

Generally speaking, unsupervised models are studied to ease the requirements for expensive human annotations. However, the unsupervised coreference models have not yet surpassed the supervised ones. In this section, an unsupervised clustering method, three unsupervised probabilistic models and one bootstrapping method for coreference resolution are described.

2.3.1 Cardie and Wagstaff's Clustering Method

Cardie & Wagstaff (1999) represent mentions to be resolved as vertices in a graph. Edge weights are calculated from a distance metric which measures the compatibility degree between vertices. The proposed distance metric is

$$dist(NP_i, NP_j) = \sum_f w_f \times incompatibility_f(NP_i, NP_j)$$

where f corresponds to a specific pairwise feature. To generate the coreference sets, an agglomerative clustering algorithm is applied afterward, which merges compatible partial clusters according to the judgments from the distance metric. The algorithm performs in a greedy manner and does not allow clusters with incompatible mentions. Therefore it may become problematic when dealing with noisy data sets.

2.3.2 Haghighi and Klein's Bayesian Model

Haghighi & Klein (2007) propose a fully generative model for coreference resolution. A non-parametric Bayesian model is adopted in order to avoid the pre-assumption about the number of entities. For non-pronominal mentions, the model makes decisions based on their dependencies on mention heads. For pronouns, the model incorporates the parameters for the entity type, gender and number. Entity salience is added into the model too. Haghighi & Klein report higher numbers than Cardie & Wagstaff (1999) on the MUC-6 data, and show that including more unannotated data can improve the performance due to the unsupervised learning nature of their model. However, Haghighi & Klein's Bayesian model is difficult to extend, since it requires the change of the model structure to include more features.

2.3.3 Ng's EM Clustering Method

Ng (2008) recasts the unsupervised coreference resolution problem as EM clustering. The adopted joint probability is

$$P(D, C) = P(C)P(D|C)$$

where D represents an observed document and C is a clustering on it. The document is further represented by mention pairs and 7 features are applied to each pair of mentions. Therefore $P(D|C)$ is given by

$$P(D|C) = \prod_{m_{ij} \in Pairs(D)} P(m_{ij}^1, \cdots, m_{ij}^7 | C_{ij})$$

The parameters (i.e. the probabilities of the features given the clusterings) are estimated using an EM algorithm and at the end a converged clustering C is induced for each document. In order to cope with the number of possible clusterings which are exponential to the number of mentions in a document, complex schemes are proposed to choose only the best clusterings at each iteration.

Ng achieves better performance compared with the enhanced version of Haghighi & Klein's system but his system is still not comparable to supervised coreference models.

2.3.4 Poon and Domingos' Markov Logic Model

In order to perform a joint inference across mentions as opposed to focus on pairwise relations, Poon & Domingos (2008) make use of the expressive power of Markov Logic to represent relations between mentions in first-order logic. Poon & Domingos propose an unsupervised system based on Markov Logic Networks to infer the coreference sets.

Several relational features are adopted, where m stands for a mention, c for a cluster and e for an entity.

1. **Head Match for Non-pronouns:**

   $\neg IsPrn(m) \wedge InCluster(m, +c) \wedge Head(m, +t)$

2. **Mention Types Agreement:**

   $InCluster(m, c) \Rightarrow (Type(m, e) \leftrightarrow Type(c, e))$

3. **Pronoun-Cluster Types Agreement:**

   $IsPrn(m) \wedge InCluster(m, c) \wedge Head(m, +t) \wedge Type(c, +e)$

4. **Apposition Constraint:**

   $Appo(x, y) \Rightarrow (InCluster(x, c) \leftrightarrow InCluster(y, c))$

5. **Predicate Nominative Constraint:**

   $PredNom(x, y) \Rightarrow (InCluster(x, c) \leftrightarrow InCluster(y, c))$

Poon & Domingos report competitive performance of their system, benefiting from leveraging relations between mentions from the cluster-level perspective. Markov Logic provides an easy way for incorporating **cluster-level features**, which is non-trivial for pair-wise models. However, their big gain by adding appositive and predicate nominative constraints cannot be reproduced for other data sets where these relations are not taken as being coreferent.

2.3.5 Kobdani et al.'s Bootstrapping Model

Kobdani et al. (2011) collect word associations from large unlabeled data sets, and propose an unsupervised system to learn the association scores between mentions. For the testing phase, the word association scores are used in the same way as the coreference probabilities. Built upon the predictions of the unsupervised system, a self-training scheme is adopted to learn the coreference relation in a conventional supervised manner. Since no manually labeled data is used, the self-training system can be viewed as unsupervised too, and it outperforms several strong unsupervised systems.

2.4 Weakly Supervised Coreference Models

Weakly (semi-) supervised learning algorithms work with little labeled data and attempt to make use of the unlabeled data while processing. They are expected to perform better than the unsupervised methods due to the available (although limited) guidance from the training labels. In this section, several weakly supervised coreference models are described.

2.4.1 Multi-view Co-training Models

Co-training (Blum & Mitchell, 1998) is a multi-view method to bootstrap by gradually extending the training (labeled) set with the *automatically* labeled data. Co-training algorithms utilize multiple learners each of which captures a separate view of the data (i.e. using disjoint subsets of features to represent the data).

Müller et al. (2002) apply a co-training method to coreference resolution by using two classifiers and therefore two views of the data. They propose a feature selection strategy to create the two subsets of features, representing the two views with the two best features and selecting the remaining one by one. Besides the greedy feature selection method by Müller et al., Ng & Cardie (2003) also experiment with random selection and the selection according to the feature types. The two classifiers are trained with their own feature sets, and predict labels for the unlabeled data. At each iteration of training, each classifier chooses its most confident predictions and add the auto-labeled data into the training set of the other classifier.

However, the results reported by Müller et al. are mostly negative and Ng & Cardie do not generate improvements with co-training algorithms either. The main difficulties lie in the generation of the independent feature sets (views), the choice of the number of iterations and the training data growth speed (Pierce & Cardie, 2001).

Raghavan et al. (2012) propose semantic and temporal features as views for their co-training classifiers, and these views appear to work on clinical data sets.

2.4.2 Single-view Bootstrapping Methods

Ng & Cardie (2003) compare multi-view weakly supervised methods with single-view ones with the application to coreference resolution. They propose two single-view algorithms, a self-training algorithm and an EM algorithm. Both of their single-view methods are based on the bootstrapping scheme.

The self-training algorithm involves a committee of classifiers, each of which is trained on a random sampled subset of the labeled data. The classifiers predict for all the unlabeled data and the predictions agreed by all of the classifiers are added to the labeled data.

The single-view weakly supervised EM assumes a parametric model of data generation. The unlabeled data are considered to be missing labels and the algorithm optimizes the posterior probability of the parameters given both the labeled and the unlabeled data. More details can be found in Nigam et al. (2000).

Ng & Cardie (2003) conclude that the single-view methods easily outperform the multi-view co-training algorithm for the coreference resolution task.

2.5 Supervised Coreference Models

Due to the existence of well-annotated corpora (see Chapter 3 for details), more attention has been paid recently to supervised coreference resolution modeling. Although coreference resolution is a *set* problem (i.e. grouping mentions into sets), the first machine-learning-based approach applies pairwise classification models which break down the problem into two-step processing (Section 2.5.1). The success of the two-step method is mainly due to its expressive simplicity and straightforward learning strategy. However, more global models are coming into the field (Section 2.5.3) aiming to conquer the performance bottle-neck from the missing of pairwise-beyond information (e.g. relations between more than two mentions).

Both local and global models are introduced in this section, so that readers can grasp an idea of the motivations and the importance of working on global models, specifically on the relative simpler graph-partitioning-based inference.

2.5.1 Two-step Methods

The Mention-pair model was firstly proposed by Aone & Bennett (1995) and McCarthy & Lehnert (1995). However, Soon et al.'s system (Soon et al., 2001) is the first successful attempt applying machine learning technique to the mention-pair model for coreference resolution, which has become the most widely used baseline system in the field.

Soon et al. divide the task into a two-step processing, a classification step and a clustering step. In step 1, the classifiers perform on pairs of mentions to decide whether they are coreferent or not. Based on the classification decisions, the clustering component merges mention pairs into sets so that all mentions in one set are coreferent to each other. A decision tree classifier (e.g. C5 Quinlan (1993)) is adopted along with 12 features for step 1, and the closest-first search strategy for step 2 (i.e. choosing the closest positive antecedent for the focusing anaphor). A simple example illustrating the two-step processing is given below.

- **Mention list**:

  a_1, b_1, a_2, b_2, a_3

- **Step 1: Classification step**:

  For b_1: $a_1 \leftarrow| b_1$

  For a_2: $a_1 \leftarrow a_2$; $b_1 \leftarrow| a_2$

  For b_2: $a_1 \leftarrow| b_2$; $b_1 \leftarrow b_2$; $a_2 \leftarrow| b_2$

  For a_3: $a_1 \leftarrow a_3$; $b_1 \leftarrow| a_3$; $a_2 \leftarrow a_3$; $b_2 \leftarrow| a_3$

- **Step 2: Clustering step**:

Set_1: $\{a_1, a_2, a_3\}$

Set_2: $\{b_1, b_2\}$

The sign $\leftarrow|$ denotes that the mention pair is decided not to be coreferent and the sign $\leftarrow$ applies to the ones which are predicted to be coreferent with each other.

In the literature, one line of improvements after Soon et al. is made along two directions, either by proposing more powerful pairwise classifiers (in step 1) or by clustering the pairwise decisions with better algorithms (in step 2). For a more detailed overview, readers are referred to Ng (2010).

Work on the Classification Step. Step 1 can be improved by exploring more powerful classifiers. Besides the decision tree classifier (e.g. Soon et al. (2001), Ng & Cardie (2002)), the Maximum Entropy classifier (e.g. Luo et al. (2004)) and the averaged perceptron learning algorithm (e.g. Bengtson & Roth (2008)) have also been applied to the classification step.

There have been researchers working on enriching the feature set for step 1. Ng & Cardie (2002) extend Soon et al.'s feature set to a size of 52, including more sophisticated linguistic knowledge. Bengtson & Roth (2008) stress on the importance of feature selection and propose to serve as the enhanced baseline system for complex coreference models.

Ponzetto & Strube (2006) firstly exploit semantic features (by the means of semantic role labeling) and world knowledge (from Wikipedia) for coreference resolution, and Rahman & Ng (2011) proceed to analyze in details the behavior of combining world knowledge with different models. Since world knowledge (especially when obtained from the web data) is noisy, it is still of interest how to make use of it in a robust way. More recent attempts can be found in Kobdani et al. (2011) and Bansal & Klein (2012).

Work on the Clustering Step. By always choosing the closest positive antecedents (as in Soon et al. (2001)), the pairwise decisions from the classification step are linked into sets. Since the *closest-first* strategy is too sensitive to error propagation, a *best-first* method is proposed by Ng & Cardie (2002) instead to link the most confident positive antecedents.

Luo et al. (2004) perform a greedy search on a bell tree representation (Figure 2.1). In each step a decision is made to connect a focusing anaphor (e.g. 3*) with a previously constructed partial entity (e.g. [12]). Although this method moves towards entity-level modeling, the greedy (and sequential) nature of the algorithm excludes important information contained in all the other paths except for the chosen one.

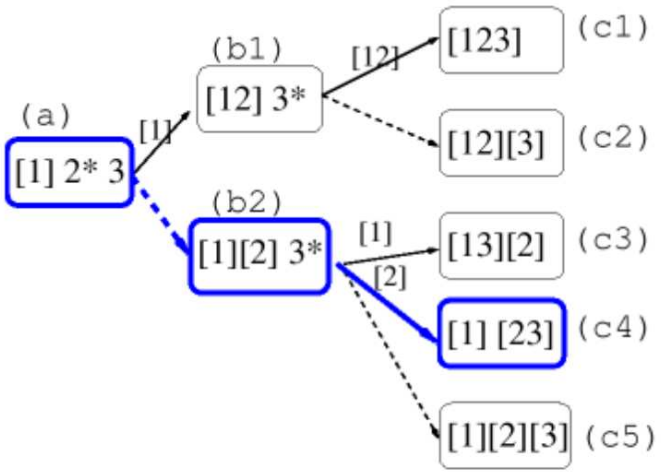

Figure 2.1: Luo's Bell Tree Method (Luo et al., 2004)

Optimization algorithms have been applied to the clustering step, in order to achieve better performance given the output from the classification step. For instance, both Klenner (2007) and Finkel & Manning (2008) impose transitivity constraints on integer linear programming (ILP) to enforce transitive closure which cannot be taken care of by greedy algorithms.

2.5.2 Preference Models

Selecting the correct antecedent for an anaphor among all candidate antecedents can also be approached by preference modeling, which predicts the winning candidates based on comparisons between all candidates. Preference models allow one to consider not only the coreference relations between antecedents and anaphors, but also the competition relation between antecedents.

Twin Candidate Model. A twin candidate model is proposed by Yang et al. (2005) to model the competition between pairs of antecedents. Each anaphor ana together with two candidate antecedents $ante_1$ and $ante_2$ form one tuple instance $\{ana, ante_1, ante_2\}$, which has three possible labels — 10 suggesting the preference of $ante_1$, 01 suggesting $ante_2$ and 00 indicating ana being non-anaphoric. The best antecedents are ranked top in a round-robin manner.

Yang et al. propose features describing relations between a pair of antecedents, which are not accessible for non-preference models.

- inter_SentDist: Distance between $ante_1$ and $ante_2$ in sentences

- inter_StrSim: 0,1,2 if $StrSim(ante_1, ana)$ is equal to, larger or less than $StrSim(ante_2, ana)$ (where $StrSim(\cdot, \cdot)$ measures the string similarity between two mentions)

- inter_SemSim: 0,1,2 if $SemSim(ante_1, ana)$ is equal to, larger or less than $SemSim(ante_2, ana)$ (where $SemSim(\cdot, \cdot)$ measures the semantic agreement between two mentions in WordNet)

Ranking Models. Denis & Baldridge (2007) rank all candidate antecedents for pronoun anaphors simultaneously, and the system is shown to outperform the twin candidate model significantly. To be able to exploit cluster-level information upon the mention ranking model, Rahman & Ng (2009) propose to rank clusters instead of antecedents.

The preference models start exploring the global relations without assuming pairwise predictions given. But due to their sequential property, only the preceding context of each anaphor is participating in the decision making which is still similar to the two-step methods.

2.5.3 One-step Methods

In this section, one-step models for the coreference resolution task are introduced. Those are the closest work to ours in terms of resolving all mentions simultaneously by considering the available full context.

2.5.3.1 Clustering Methods

Two algorithms are described in this section, both of which perform the global inference by means of clustering algorithms.

Nicolae and Nicolae's graph clustering algorithm to be introduced is still built upon pairwise classification output (as edge weights). However, it is considered as a global model as they do not sequentially cluster mentions into coreference sets, but resolve them all together.

Cardie and Wagstaff's Method. It is worth noting that Cardie and Wagstaff's method (Cardie & Wagstaff, 1999) in Section 2.3 is unsupervised since the edge weights are set manually. However their clustering mechanism can be easily adapted into a supervised version by learning the weights automatically.

Recall that Cardie & Wagstaff represent mentions to be resolved as vertices in the graph, and edge weights are calculated from a distance metric which measures the compatibility degree between vertices. An agglomerative clustering algorithm is applied to generate the coreference sets afterward.

Nicolae and Nicolae's Best-cut. Nicolae & Nicolae (2006) describe a graph-cut-based algorithm with the same graph representation as Cardie and Wagstaff's. The graph-cut strategy superficially resembles our approach. However, they apply the cutting algorithm only on the output from a classification step which form a weighted standard graph as shown in Figure 2.2.

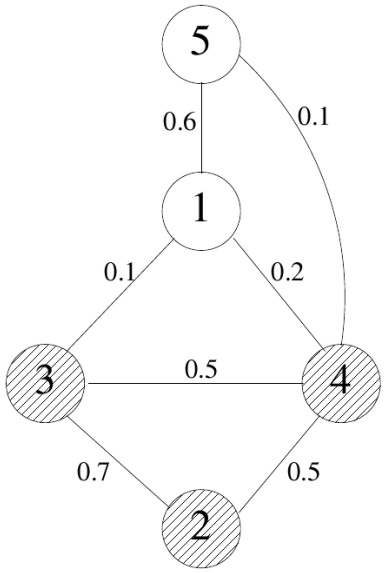

Figure 2.2: Nicolae and Nicolae's Best-cut Method (Nicolae & Nicolae, 2006)

They report considerable improvements over state-of-the-art systems including Luo et al. (2004). However, since they not only change the clustering strategy but also the features for the classification step, it is not clear whether the improvements are due to the graph-based clustering technique. Furthermore, they separate pronoun resolution from the core processing but adopt a standard two-step method for pronouns. The fact that their algorithm is only applied to a subset of mentions makes it less elegant than ours.

2.5.3.2 Probabilistic Models

Being conceptually similar to the graph clustering algorithms, probabilistic models optimize the entity assignments by considering all relations available in the focusing contexts. Different inference frameworks have been explored in the literature to capture cluster-level information (e.g. transitivity) and different approximation algorithms are used to make globally optimized predictions. It is not very clear yet which model is distinguishably superior.

McCallum and Wellner's Conditional Model. McCallum & Wellner (2005) introduce three discriminative, conditional-probabilistic models for coreference resolution, all examples of undirected graphical models. The models condition on the mentions, and generate entity assignments for them. It is shown that the most improved version (i.e. the third model) can transform itself to an equivalent (different) graph, which is with mentions as vertices and edge weights ranging from $-\infty$ to $+\infty$. The inference thus becomes a graph partitioning problem, where e.g. correlation clustering (Bansal et al., 2002) can be applied to handle the negative edges.

Culotta's First-order Logic Method. Culotta et al. (2007) adopt a first-order logic representation where features over sets of mentions are implemented (i.e. cluster-level features). The proposed models can be viewed as estimating the parameters for each cluster-wise compatibility independently and then being combined together via clustering. Uniform sampling is used for generating training instances (i.e. positive/negative clusters) in one model, and on-line training schemes are proposed for the other two improved versions. They use four features in the model. The first is an enumeration over *pairs* of noun phrases. The second is the output of a *pairwise* model. The third is the cluster size. The fourth counts mention type, number and gender in each cluster. They assume true mentions as input and only report one evaluation metric numbers. It is not clear whether the improvement in results translates into system mentions.

Sapena's Relaxation Labeling Algorithm. Sapena et al. (2010) use a constraint-based approach (i.e. relaxation labeling) for coreference resolution. They generate pairwise predictions as constraints using a decision tree classifier and represent them in a graph. Afterward they optimize with respect to the constraints (both positive and negative ones) in an iterative procedure. It is shown that the proposed model outperforms an ILP algorithm with the transitivity enforced.

In his thesis (Sapena, 2012), Sapena shows that his graph representation can be viewed as hypergraphs, as illustrated in Figure 2.3. The mentions are taken as vertices and the constraints generated from the decision tree are taken as edges (e.g. e_1, e_2 and e_3). The main differences between Sapena's work and ours lie in (1) his hyperedges represent the learned combinations of features while ours are derived directly from simple (low-dimensional) relational features; (2) his resolution model is a probabilistic model while ours performs under the graph-based clustering framework. The two work differs in both the representation model and the resolution algorithm, despite of the similar namings.

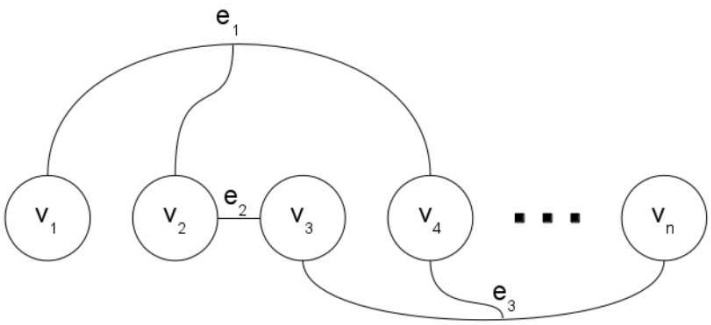

Figure 2.3: Sapena Thesis's Hypergraph Representation (Sapena, 2012)

Markov Logic Models for Coreference Resolution. As mentioned, Poon & Domingos (2008) propose to use a learning-based unsupervised Markov Logic Model for coreference resolution, which manages to incorporate cluster-level features via formulas. Song et al. (2012) implement a supervised framework using Markov Logic, to perform the mention pair classification and the mention clustering jointly. They make use of the expressive power of Markov Logic Networks to include hard (global) constraints for the *best-first* scheme and for transitivity. Frank et al. (2012) adopt Markov Logic Networks to detect errors in automatic semantic annotations. The automatic system predictions for word sense disambiguation and coreference resolution are taken together into the their model, and are optimized (i.e. corrected) via the joint inference. Both Song et al.'s and Frank et al.'s proposed models can be viewed as optimization methods for step 2 in the two-step coreference framework.

2.6 Summary

Two-step Coreference Models. Although coreference resolution is naturally a clustering problem, which aims to cluster mentions into coreference sets, most of the recent approaches divide the task into two steps: (1) a classification step which determines whether a pair of mentions is coreferent or which outputs a confidence value, and (2) a clustering step which groups mentions into entities based on the output of step 1.

Soon et al. (2001) firstly propose the two-step strategy under the machine learning framework, i.e. pairwise classification and clustering. They use a set of twelve powerful features. Their system is based solely on information of the mention pairs (i.e. anaphor and antecedent), and does not take any information of other mentions into account. However, it turned out that it is difficult to improve upon their results by just applying a more sophisticated learning method without improving the features.

A number of approaches have been focusing on improving coreference modeling within the two-step framework, either by proposing linguistic-leaned or world-knowledge-based features or by applying different optimization algorithms for the clustering phase. Most of the two-step methods are considered to be local, because they make coreference decisions on pairs of mentions and cluster the mentions into sets considering only the preceding antecedents. In order to exploit the full context, **global models are preferred over the two-step methods**.

Global Coreference Models. As an example of graph partitioning models for coreference resolution, Nicolae & Nicolae (2006) propose a graph-cut-based approach where mentions are vertices and edge weights are learned from pairwise coreference classifiers. Unfortunately, they only manage to resolve non-pronoun mentions in this framework and have to approach pronoun resolution separately. This work is superficially similar to ours, but our graph-based model includes mentions of all types in the graph representation. In this way, we are able to access the full context of the focusing document, which makes our model fully global.

Graphical models have the superiority of precise probability formulating, which consequently enables the coreference systems to learn complex dependency structures between mentions and entities. However, the learning and inference procedures can be complicated even with the approximation (e.g. Finkel & Manning (2008)), which make them less preferable than the simpler coreference systems such as ours.

Lang et al. (2009) propose an unsupervised coreference resolution system based on a hypergraph partitioning algorithm, which did not appear accessible before our first proposal (Cai & Strube, 2010a). Lang et al. represent mentions as vertices and generate hyperedges directly from features. Unfortunately, no strict experimental comparison (with the same feature sets) is provided to verify the effect of their model. Furthermore, the mentions along with their heads and semantic types are all taken from the gold annotation in Lang et al.'s system.

In contrast, in this thesis we present a complete hypergraph partitioning model for coreference resolution and provide thorough experiments with realistic system settings. Crucial issues regarding both the clustering algorithms and the coreference application are addressed in this thesis. For instance, we propose the feature categorization in Chapter 5 to ensure the stable construction of the hypergraphs. Extensive experiments across different domains and different evaluation metrics are able to convey the effectiveness and the robustness of our proposed system.

Chapter 3

Data Sets for Coreference Resolution

Two data sets have been frequently used for years to evaluate coreference resolution. The former is from the MUC conferences (see Section 3.1) and the latter is provided by the Automatic Content Evaluation (ACE) program (see Section 3.2). Stoyanov et al. (2009) point out that there are significant differences in annotating mentions and the coreference relation between these corpora, which will be illustrated in this chapter. A much larger corpus OntoNotes (see Section 3.3) was recently released. It became the standard evaluation set for the coreference resolution task soon after its first usage in the CoNLL 2011 shared task (Pradhan et al., 2011).

In this thesis, we also experiment on a medical data set (see Section 3.4), which consists of clinical reports with annotated coreference relation between persons, (clinical) problems, treatments etc.

We describe the coreference data sets before introducing our proposed coreference model in this thesis, aiming to assist the readers to better understand coreference phenomena and the annotation- scheme-related problems involved in the task.

3.1 MUC

The MUC data sets consist of MUC-6 (MUC-VI Text Collection) (Chinchor & Sundheim, 2003) with a standard training/testing division (30/30) and MUC-7 data (North American News Text Corpora) (Chinchor, 2001) (30/20). The documents in the MUC data sets are all news articles, and are prepared (annotated) for four evaluation tasks — Named Entity Recognition, Coreference Resolution, Template Elements and Scenario Templates.

The MUC corpora are annotated with general types of mentions, but only the ones that participate in the coreference relation. In other words, the entities containing single mentions (denoted as *singleton entities*) are not tagged, such as ”the Federal Railway Labor Act” in the following MUC Example. It is also worth noting that neither apposition nor predicate nominatives are annotated as the coreference relation.

MUC Example:

Under the Federal Railway Labor Act, if the mediator fails to bring [*the two sides*]$_1$ together and [*the two sides*]$_1$ do n't agree to binding arbitration, [*a 30-day cooling-off period*]$_2$ follows .

After [*that*]$_2$, [*the union*]$_3$ can strike or the company can lock [*the union*]$_3$ out .

Since we only focus on the end-to-end coreference resolution problem, which takes raw texts as input without assuming any annotations, mentions need to be detected automatically. Our mention tagger (see Chapter 7) tends to identify too many mentions for MUC data, as there is no restriction on the types of mentions to be resolved. This is therefore resulting in too many spurious coreference sets, such as the entity containing several [*yesterday*] mentions.

3.2 ACE

There are four corpora from the ACE program, ACE 2002 (Mitchell et al., 2002), ACE 2003 (Mitchell et al., 2003), ACE 2004 (Mitchell et al., 2004) and ACE2005. The annotations of ACE data contain six areas — Entity Detection and Recognition (EDR), Entity Mention Detection (EMD), EDR Co-reference, Relation Detection and Recognition (RDR), Relation Mention Detection (RMD), and RDR given reference entities. There are different types of document sources for ACE data sets, i.e. news wire reports, broadcast news programs and newspapers, and in three different languages, i.e. Arabic, Chinese and English. In this thesis, we use both ACE 2003 and ACE 2004. Since we do not have access to official ACE testing data (only available to ACE participants), we follow Bengtson & Roth (2008) to divide ACE 2004 English training data into training, development and testing partitions (268/76/107). We randomly split the 252 ACE 2003 training documents using the same proportions into training, development and testing (151/38/63).

The coreference relation in ACE data sets is annotated only among the mentions of certain entity types. For instance, ACE 2004 adopts 7 entity types, which are Person (PER), Organization (ORG), Location (LOC), Geo-Political Entity (GPE), Facility (FAC), Vehicle (VEH) and Weapon (WEA). Singleton entities are allowed in ACE data as long as they are of the required entity types. In the following ACE Example that illustrates the ACE annotations, both mentions [*Palestinian*]$_1$ and [*the former Soviet Union*]$_4$ form singleton entities due to their GPE types.

ACE Example:

> The problem arose after [[*Palestinian]*$_1$ *Mahmood Abu Talib,* [*whose*]$_2$ *testimony the court has been hearing since Friday*]$_2$, refused to continue answering a question by [[*defense lawyer*]$_3$ *Richard Keen*]$_3$ about the detailed reasons for [*his*]$_3$ having lived in [*the former Soviet Union*]$_4$ for a period of 18 months in the 70s .
>
> [*The lawyer*]$_3$ asked the judges to force [*Abu Talib*]$_5$ to answer the question aimed at demonstrating [*the witness*]$_5$'s "professional terrorism" precedents .

There are several special relations that are taken as the coreference relation in ACE data sets, such as appositive (e.g. entity 2), predicative nominative and role appositive (e.g. [[*defense lawyer*]$_3$ *Richard Keen*]$_3$). Features designed for capturing these special relations might not work when moving to different data sets, as they usually do not form the coreference relation from the linguistic perspectives.

It is relatively easier to detect ACE mentions given the fixed entity types. However, since entity extraction is also implicitly evaluated via singleton entities, it brings non-trivial implementation issues to the the coreference evaluation metrics (for more details, readers are referred to Chapter 6).

3.3 OntoNotes

The OntoNotes Release 4.0 corpus (Weischedel et al., 2011) provided by the Linguistic Data Consortium (LDC) is used for CoNLL 2011 shared task on modeling unrestricted coreference in OntoNotes. It consists of $2,999$ English documents, $1,674$ of which are chosen as the training data, 202 as the development set and 207 as the testing set for the shared task. In the collection, there are news wire texts, broadcast news, broadcast conversations, magazine and web documents. The diverse text types impose more challenges on coreference systems.

In addition to the coreference relation, OntoNotes data is also tagged with syntactic trees, high-coverage verb and some noun propositions, partial verb and noun word senses, and 18 named entity types. The shared task provides two types of annotation layers, the gold layers (for the training set) and the system predicted layers (for all sets). The participating systems can only have access to system predicted information during the testing phase, which explicitly stresses on the importance of the end-to-end coreference setting.

In OntoNotes data, appositive structures are annotated as a separate type and they are not included in the coreference sets. The predicative nominatives are not considered being coreferent either. Event coreference is annotated, such as the [*overcoming*]$_2$ and [*This example*]$_2$ entity in the following OntoNotes Example (1). As shown in OntoNotes Example (2), the generic phrases (e.g. [*Officials*]$_1$) are also tagged as mentions as long as there are other men-

tions being coreferent with them. GPEs are linked to the references of their governments, e.g. [*China*]$_1$ and [*the Chinese government 's*]$_1$ in OntoNotes Example (3).

OntoNotes Example (1):

[*The South Korean team of veterans*]$_1$, by [*overcoming*]$_2$ [*their*]$_1$ injuries to give a display of athleticism at the international level , have emerged from the shadow of war and transformed [*their*]$_1$ handicaps into glorious results.

[*This example*]$_2$ should provide food for thought to the disabled and sports communities in the future .

OntoNotes Example (2):

[*Officials*]$_1$ say [*they*]$_1$ have reduced the reunion schedule from four days to three and will spend some $ 800,000 to bring the families together , compared with the nearly $ 1.6 million it spent for the August event .

OntoNotes Example (3):

[*China*]$_1$ today blacked out a CNN interview that was critical of [*the Chinese government 's*]$_1$ handling of the SARs epidemic and of [*the country 's*]$_1$ health care system.

3.4 I2B2

The I2B2/VA/Cincinnati Childrens 2011 challenge (Uzuner et al., 2012) held one NLP shared task in 2011, the first track of which was on coreference resolution. Participants were asked to mark the concept mentions (i.e. entity mentions), including pronouns, as coreferent or not. Data for this track were provided by *Partners HealthCare*, *Beth Israel Deaconess Medical Center* (MIMIC II Database), *University of Pittsburgh*, and *the Mayo Clinic*. According to different settings, the task was further divided into task 1A, 1B and 1C. We participated in all three of them.

The ODIE corpus (including the Mayo and Pittsburgh data sets) is used for task 1A and task 1B. Task 1B provides manually annotated mentions (referred to as concepts in the task description) while task 1A requires an automatic mention detection. The ODIE corpus consists of 97 training documents. The I2b2/VA/Cincinnati corpus (including the *Partner*, *Beth* and

Pittsburgh data sets) with 492 training documents is used for task 1C, where the true mentions are provided too.

The entities of interest in the I2B2 data sets are significantly different from the ones in standard coreference data sets (i.e. the previously introduced corpora in this chapter), which cover persons, problems, treatments, tests, etc. All the texts are in semi-structured formats, with content of the clinical treatments a patient receives as well as a rich set of his/her relevant information, e.g. the admission date, the date of birth, etc.

I2B2 Example (1):

[*Attending*]$_1$:

[*Gayle M Whitener , M.D.*]$_1$

I2B2 Example (2):

On hospital day 2 she experienced [*atrial fibrillation*]$_1$ with HR in the 140s.

We decided given her age that she would not be a good candidate for cardioversion for [*her afib*]$_1$ nor would she be a good candidate for coumadin.

I2B2 Example (3):

[*VULVAR CANCER*]$_1$.

A tumor was noted on her vulva which was biopsied and revealed [*squamous cell carcinoma in situ*]$_1$.

Examples from I2B2 corpora are shown above. It can be seen that due to the organized structures, some of the coreference entities are obvious to solve, e.g. [*Attending*]$_1$ and [*Gayle M Whitener , M.D.*]$_1$ in I2B2 Example (1). However, abbreviations (e.g. [*atrial fibrillation*]$_1$ and [*her afib*]$_1$ in I2B2 Example (2)) can be difficult as well as the variants for medical expressions (e.g [*VULVAR CANCER*]$_1$ and [*squamous cell carcinoma in situ]*$_1$ in I2B2 Example (3)).

3.5 Summary

In order to convey the improvements one achieves, researchers in the corefernce resolution field always conduct comparison experiments on several standard data sets. The documents selected for the corpora are conventionally news articles. The community starts to include

speech transcripts and others only recently in OntoNotes data. In this chapter, the coreference data sets used by our system are introduced, including one additional medical corpus.

The given examples show that the entity types and the annotation schemes vary between different data sets, so that the corpus-specific system engineering and feature designing are necessary to some degree. For instance, features capturing the knowledge on GPE entities are required for news articles, while for clinical reports, medical-domain-specific knowledge are needed in order to solve the difficult cases. Nevertheless, linguistically driven features (e.g. binding constraints) can be applied universally.

Chapter 4

COPA: Coreference Partitioner

In this thesis, we propose a novel coreference resolution model, that represents documents as hypergraphs, upon which partitioning algorithms are applied to derive the coreference sets directly and simultaneously. Our system is named ***COPA***, standing for Coreference Partitioner.

The Hypergraph Representation. Unlike most of the previous work that resolves the pairwise relations independently (e.g. the two-step methods in Chapter 2), representing documents as graphs enables *COPA* to have a global view of the relations between all mentions. More specifically, we propose the hypergraph model for the representation, motivated by the *high-dimension* property of the coreference relation. The standard graph models have to collapse the multiple low-dimensional relations between mentions into single ones (i.e. the coreference relation) as edges, which leads to a loss of information before the inference phase. In contrast, a hypergraph is a graph in which (a) a hyperedge can connect more than two vertices, and (b) between two vertices there can exist more than two hyperedges. Therefore, our hypergraph model is able to **maintain the original low-dimensional relations** as overlapping hyperedges (i.e. (b)) until the final inference, and the model also **easily represents sets of mentions** (i.e. (a)) which suits well the set property of coreference resolution.

The Partitioning Inference. Upon the hypergraph representation, *COPA* produces the coreference sets so that the mentions within the same sets are closely connected and different sets are far apart from each other. In order to achieve such an optimization, we propose to apply the graph partitioning technique as the inference method for coreference resolution. Graph partitioning algorithms seek for a cut upon the graph edges, so that the derived subgraphs are optimized with respect to a specific graph cut function. In *COPA*, we adopt the *Normalized Cut* (NCut) function which measures both the inner-set and the inter-set connectivities. The spectral clustering algorithm is employed to optimize the NCut value, so that the inner-set connections are as strong as possible while the inter-set ones are as weak as possible. With the

graph partitioning algorithm applied, the **optimized coreference sets** are able to be derived **simultaneously**.

The Chapter Organization. Section 4.1 illustrates how *COPA* works via examples. The mathematical background of both the hypergraph model and the spectral clustering algorithm is described in Section 4.2, which provides the notation used throughout the thesis. Section 4.3 describes in detail our proposed hypergraph partitioning model for coreference resolution. The important issues regarding applying the graph partitioning technique to practical uses are discussed in Section 4.4. As mentioned previously, the hypergraph is a generalization of the standard graph and is equipped with additional power of representation. However, there exist standard graphs to which the hypergraph can be transformed (see Section 4.5). Upon the standard graphs, more graph-based algorithms can be directly applied. Therefore such transformation gives the freedom in choosing the inference algorithm to hypergraph-based models. Although *COPA* performs directly on the hypergraphs, future extensions on the inference method may benefit from such graph transformation.

4.1 Introduction to *COPA*

Figure 4.1 shows the modules of our proposed coreference resolution system. The *COPA* system includes the learning modules for collecting the hyperedge weights (i.e. the *Hyperedge Learner* in Section 4.3.2) and for predicting the number of entities k (i.e. the k *model* in Section 4.3.4). The resolution modules of the *COPA* system construct the hypergraph models for the testing documents (using the *Hypergraph Builder* in Section 4.3.2) and partition them into sub-hypergraphs (using the *Hypergraph Resolver* in Section 4.3.3).

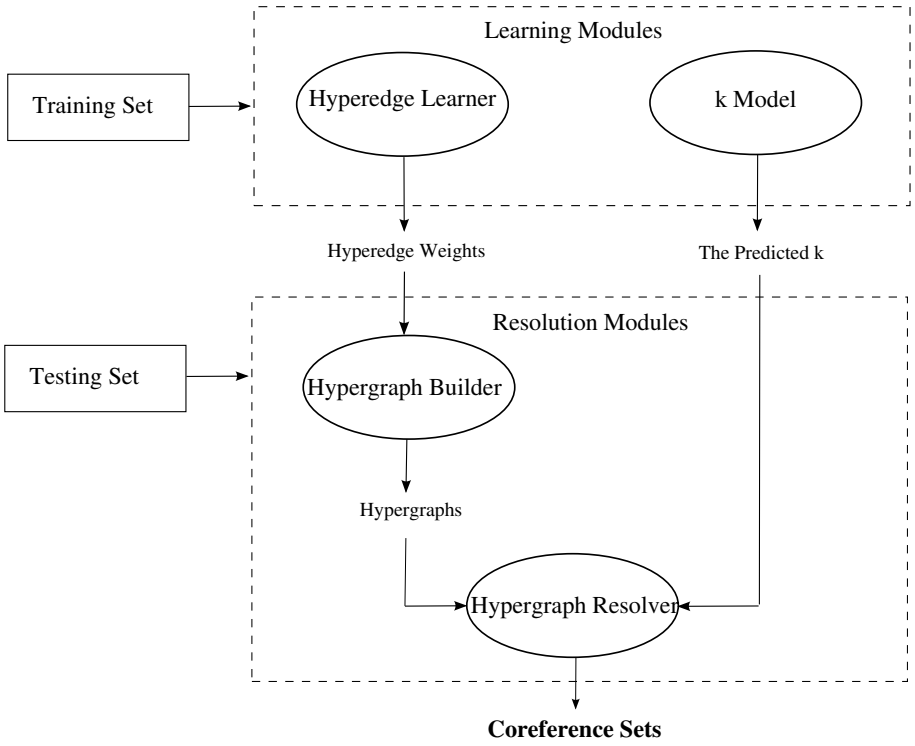

Figure 4.1: COPA Model Illustration

***COPA* Example.** To illustrate how *COPA* works, an example of a short document involving two entities — BARACK OBAMA and NICOLAS SARKOZY — is provided in Table 4.1.

[US President Barack Obama] came to Toronto today.
[Obama] discussed the financial crisis with [President Sarkozy].
[He] talked to [him] about the recent downturn of the European markets.
[Barack Obama] will leave Toronto tomorrow.

Table 4.1: *COPA* Example: Texts

A hypergraph (Figure 4.2 a) is built for the example document based on three features. Two red (solid line) hyperedges denote the feature *partial string match* — {*US President Barack Obama, Barack Obama, Obama*} and {*US President Barack Obama, President Sarkozy*}. One green (dashed line) hyperedge denotes the feature *pronoun match* — {*he, him*}. Two blue (dashed-dotted line) hyperedges denote the feature *subject*|*object match* — {*Obama, he*} and {*President Sarkozy, him*}. Each of the hyperedges has an associated edge weights (the examples of which can be seen in Section 4.3.2).

On this initial representation, spectral clustering technique is applied to find two partitions that have the strongest within-cluster connections and at the same time the weakest between-clusters relations. The cut found in this way is called *Normalized Cut* (abbreviated as *NCut*), which avoids trivial partitions frequently output by the min-cut algorithm (see Section 4.2.2). The two resulting sub-hypergraphs (Figure 4.2 b) correspond to two resolved entities shown on both sides of the bold dashed line, i.e. the upper left sub-graph being BARACK OBAMA and the lower right NICOLAS SARKOZY. In real cases, multiple entities can be found within one document.

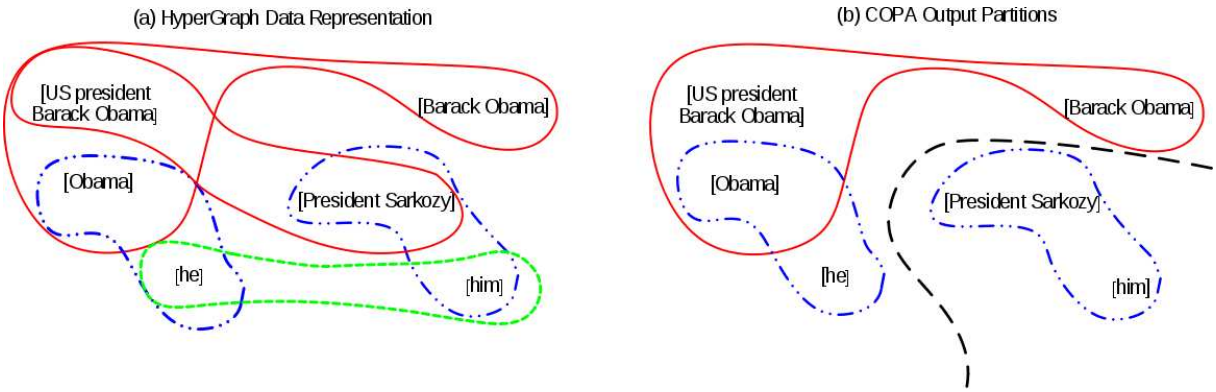

Figure 4.2: COPA Example: Processing Illustration

4.2 The Mathematical Background

4.2.1 The Hypergraph Representation

A hypergraph is a graph in which hyperedges can connect more than two vertices, and between two vertices there can be multiple hyperedges.

The Hypergraph Notation. Let $HG = (V, E)$ be a hypergraph with a vertex set V and a hyperedge set E. The hyperedges can connect arbitrarily multiple vertices such that $E \subseteq \{U|U \subseteq V, |U| > 1\}$. A weighted HG has a positive weight value $w(e)$ associated with each hyperedge e. A vertex v is incident with a hyperedge e if it is connected with the edge, being denoted as $v \in e$.

For a vertex $v \in V$, the degree of v is the number of hyperedges connecting to it and is thus defined as

$$d(v) = \sum_{e \in E | v \in e} w(e) \tag{4.1}$$

For a hyperedge $e \in E$, its degree is the number of vertices connected by it, denoted as

$$\delta(e) = |e| \tag{4.2}$$

In order to be analyzed mathematically, the hypergraph representation is further transformed into matrices. The incidence matrix H of a HG is a $|V| \times |E|$ matrix with entries $H(v, e) = 1$ if $v \in e$ and 0 otherwise. D_v and D_e denote the diagonal matrices with the vertex and hyperedge degrees respectively, and W the diagonal matrix with the corresponding hyperedge weights. After the transformation, the matrices contain full information about the original hypergraphs.

The Matrix Computation Example. We use the hypergraph in Figure 4.3 as an example to illustrate the matrix computations introduced above. The numbers in brackets are the corresponding hyperedge weights.

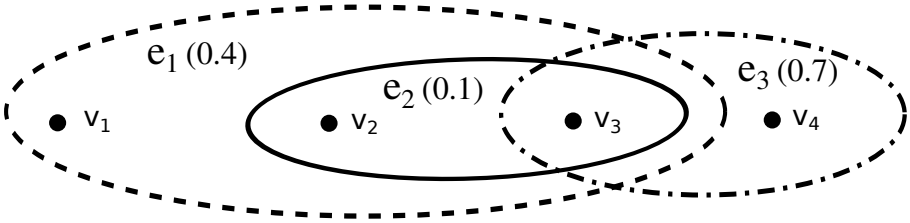

Figure 4.3: An Example for the Hypergraph Notation

The incidence matrix H of this hypergraph and the hyperedge weight matrix W are

$$H = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}, W = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{pmatrix} 0.4 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.7 \end{pmatrix} \end{array}$$

The degrees of vertices are calculated as

$$d(v_1) = w(e_1) = 0.4$$
$$d(v_2) = w(e_1) + w(e_2) = 0.5$$
$$d(v_3) = w(e_1) + w(e_2) + w(e_3) = 1.2$$
$$d(v_4) = w(e_3) = 0.7$$

so that the vertex degree matrix D_v and the hyperedge degree matrix D_e are

$$D_v = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0.7 \end{pmatrix} \end{array}, \quad D_e = \begin{array}{c} \\ e_1 \\ e_2 \\ e_3 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \end{array}$$

4.2.2 Hypergraph Partitioning

Grouping data into meaningful clusters is well known as *cluster analysis* or *data clustering*, which is to discover the intrinsic structures of the focusing data sets (see Jain et al. (1999) for an overview). The data points to be clustered are usually in vector-based feature representations, the quality of which often influences the performance of the clustering algorithms directly. For tasks where the relations between data points are of greater interest, such as coreference resolution, explicit data vector representations can be avoided by resorting to graph models.

Partitioning upon graphs is also referred as *graph clustering*. Graph clustering is the task of dividing the vertices in a graph into sets (i.e. sub-graphs), such that vertices within sets are tightly connected to each other in some pre-defined sense, while the ones from different sets are loosely related. The edges to be removed to output the sub-graphs form a *cut*, and the edges are said to be crossing the cut. In a weighted graph, the *value* of a cut is defined by the sum of the weights of these edges crossing the cut. Graph clustering algorithms are aiming at finding a partition that optimizes the chosen cut value, so that the partition provides an optimal segmentation solution on the graph.

Spectral clustering is a family of clustering algorithms that has been proven to work efficiently in applications and frequently outperforms standard clustering algorithms such as k-means. In *COPA*, we adopt a spectral clustering algorithm that can perform directly on hypergraph models.

4.2.2.1 Spectral Clustering

Taking the two-way partitioning as an example, we introduce briefly the intuitions behind spectral clustering in this section.

The Standard Graph Cut. Let A, B denote two disjoint sub-graphs from the original graph $G = (V, E)$ (V, E are vertex set and edge set respectively), where $A \cup B = V$ and $A \cap B = \emptyset$. The standard *graph cut* is defined as

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{4.3}$$

Finding the minimum cut (*min-cut*) of a graph (i.e. $min_{A,B}(cut(A, B))$) is the simplest and most direct way to solve the partitioning problem. The *min-cut* is well-studied (see Stoer & Wagner (1997) for algorithms and discussions) and is used in applications too (Wu & Leahy, 1993). However, it is noticed that the *min-cut* criteria favors cutting isolated vertices (Jain et al., 1999) , which have few edges connecting to others in the graph so that the corresponding cut value is small. Most applications focus on detecting meaningful cluster structures (i.e. the clusters consisting of multiple vertices), and are not interested in such trivial singletons output by *min-cut* algorithms.

Normalized Cut. Shi & Malik (2000) propose a new measure of disassociation between sub-graphs, taking the inner-cluster density into consideration too. The new measure is called *Normalized Cut* (*NCut*):

$$NCut(A, B) = \frac{cut(A.B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{4.4}$$

Where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ sums all the edges between vertices in A sub-graph and all vertices in the original graph. Therefore, by minimizing the *NCut* value, the resulting sub-graphs should be weakly connecting to each other while being as dense as possible at the same time.

However, introducing the inner-cluster factor makes the minimization of *NCut* an NP-hard problem. Spectral clustering techniques (Chung, 1997; Shi & Malik, 2000; Ng et al., 2002) solve the relaxed version by partitioning the rows of a matrix (see the Laplacian matrix L_{sym} in Section 4.2.2.2) according to the components in the top few singular vectors for the matrix. They are simple to implement and reasonably fast and have been shown to frequently outperform traditional clustering algorithms such as k-means algorithm in applications (von Luxburg, 2007).

4.2.2.2 Spectral Clustering for Hypergraphs

(Zhou et al., 2007) generalize spectral clustering to operate directly on hypergraphs (in contrast to e.g. Agarwal et al. (2005) who partition a graph that approximates the hypergraph). In *COPA*, we adopt their hyperspectral clustering algorithm.

Following the same intuition behind the standard normalized cut as introduced in Section 4.2.2.1, hypergraph spectral clustering defines the $NCut_{hg}$ of a k-way partitioning P_k as

$$NCut_{hg}(P_k) = \sum_{1 \leq i \leq k} \frac{vol\partial V_i}{volV_i} \tag{4.5}$$

Where $V_i \cap V_j = \emptyset$, for all $1 \leq i, j \leq k$ and $i \neq j$.

The volume $volV_i$ of a vertex set V_i is defined by

$$volV_i = \sum_{v \in V_i} d(v) \tag{4.6}$$

The hyperedge boundary ∂V_i is defined as the graph cut separating V_i from other vertices in the graph, such that

$$\partial V_i = \{e \in E | e \cap V_i \neq \emptyset, e \cap V_i^c \neq \emptyset\} \tag{4.7}$$

where V_i^c denotes the complement of V_i.

The volume of the hyperedge boundary is defined by

$$vol\partial V_i = \sum_{e \in \partial V_i} w(e) \frac{|e \cap V_i||e \cap V_i^c|}{\delta(e)} \tag{4.8}$$

When a minimized $NCut(P_k)$ value is reached, the linkage between clusters is as weak as possible while it is as dense as possible within clusters. The minimization can be approached using a relaxation approach, which approximates discrete cluster memberships with continuous real numbers by solving the eigen problem of the *hypergraph Laplacian*. The symmetric Laplacian (L_{sym}) (von Luxburg, 2007) is adopted.

$$L_{sym} = I - {D_v}^{-\frac{1}{2}} H W {D_e}^{-1} H^T {D_v}^{-\frac{1}{2}} \tag{4.9}$$

Given a hypergraph *HG*, a set of matrices is generated. D_v and D_e denote the diagonal matrices containing the vertex and hyperedge degrees respectively. $|V| \times |E|$ matrix H represents the *HG* with the entries $h(v, e) = 1$ if $v \in e$ and 0 otherwise. H^T is the transpose of H. W is the diagonal matrix with the edge weights.

Let $(\lambda_i, v_i), i = 1, \ldots, n$, be the eigenvalues and the associated eigenvectors of L, where $0 \leq \lambda_1 \leq \cdots \leq \lambda_n$ and $\|v_i\| = 1$. The continuous solution to minimizing $NCut(P_k)$ is then provided by a new data representation X with lower dimensions compared with the original data dimensions:

$$X = (v_1, \cdots, v_k) \tag{4.10}$$

where X is called the k-th order **spectral embedding** of the graph. It has been shown that k is generally equal to the number of clusters (Ng et al. 2001). A standard data clustering algorithm, such as the k-means method (MacQueen, 1967), can afterward be applied to cluster the graph nodes in the new space. An illustration is given in Figure 4.4 to show how spectral clustering work on graph models.

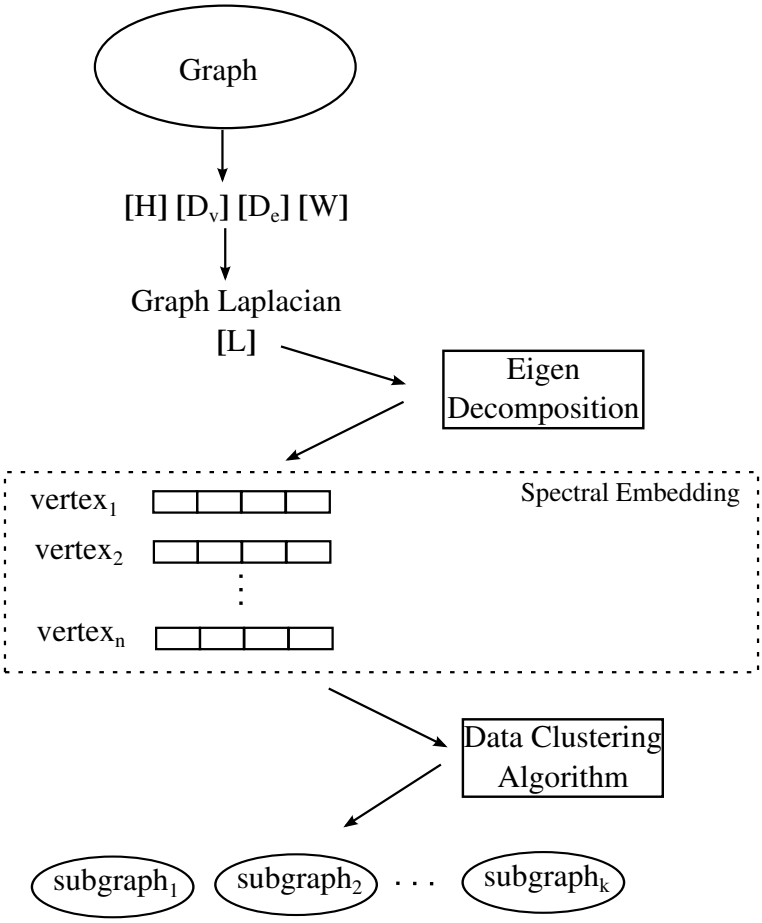

Figure 4.4: Illustration of Spectral Graph Clustering

4.3 *COPA*: Coreference Resolution via Hypergraph Partitioning

Figure 4.5 illustrates the work flow of the *COPA* system. The system takes raw documents as input and outputs the expected coreference sets. The pre-processing components perform text parsing (e.g. POS tagging and syntactic parsing), mention identification, and mention-relevant information extraction (e.g. semantic class identification). With the identified mentions and the extracted features, *COPA* represents the input text as hypergraphs. At the end, *COPA* partitions the hypergraphs into coreference sets.

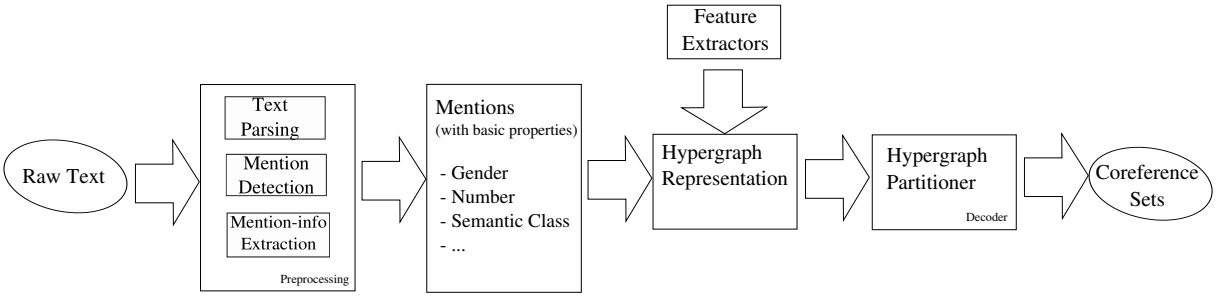

Figure 4.5: Illustration of *COPA* System Flow

4.3.1 Preprocessing Pipeline

COPA is implemented on top of the *BART*-toolkit (Versley et al., 2008). Documents are transformed into the *MMAX2*-format (Müller & Strube, 2006) which allows for easy visualization and (linguistic) debugging. Each document is stored in several XML-files representing different layers of annotations. These annotations are created by a pipeline of preprocessing components. We use the *Stanford MaxentTagger* (Toutanova et al., 2003) for part-of-speech tagging, and the *Stanford Named Entity Recognizer* (Finkel et al., 2005) for annotating named entities. In order to derive syntactic information, we use the *Charniak/Johnson reranking parser* (Charniak & Johnson, 2005) combined with a constituent-to-dependency conversion Tool [1].

We have implemented an in-house mention tagger, which makes use of the parsing output, the part-of-speech tags, as well as the chunks from the *Yamcha Chunker* (Kudoh & Matsumoto, 2000). The mention tagger detects automatically the mention boundaries, along with their syntactic heads.

The separated-annotation-layer scheme and the flexible feature representation (see Chapter 5) enable *COPA* to incorporate knowledge easily. For instance, to enrich the system with medical domain information, we query the Unified Medical Language System (UMLS) [2] and the MetaMap software (Aronson, 2001) for each mention. All the top matched concept names returned by the MetaMap API as well as their corresponding definitions in the UMLS database are collected during preprocessing.

4.3.2 Constructing Hypergraphs for Documents

The ***Hypergraph Builder*** component of *COPA* represents documents in undirected hypergraphs with basic relational features. Hyperedges are derived from the adopted feature set.

[1] `http://nlp.cs.lth.se/software/treebank_converter`
[2] `http://www.nlm.nih.gov/research/umls/`

Each hyperedge corresponds to a feature instance modeling a specific relation of that feature type between two or more mentions. This leads to initially overlapping sets of mentions (as in Figure 4.2(1a)).

Hyperedges are assigned weights that are calculated from the training data using the ***Hyperedge Learner*** component, as the percentage of the initial edges being in fact coreferent. For instance, when calculating the edge weights for the *HeadMatch* feature, 126 binary corresponding relations are found, out of which 55 are coreferent. As a result, the edge weight for *HeadMatch* is $\frac{55}{126} = 0.4365$. Since only basic statistics are collected from the annotated data, *COPA* is not sensitive to the size of the training set (see Chapter 7).

The weights for some of (Soon et al., 2001)'s features learned from the ACE 2004 training data are given in Table 4.2.

Edge Name	**Weight**
Alias	0.777
StrMatch_Pron	0.702
Appositive	0.568
StrMatch_Npron	0.657
NonPron_Pron	0.403

Table 4.2: Hyperedge Weight Examples for ACE 2004 Data

4.3.3 Hypergraph Resolver

Raw documents are transformed into hypergraphs with mentions as vertices and features as edges. In contrast to the common practice in graph models, we incorporate rich relational information directly without assuming a distance metric and maintain all the relations until the final generation of the coreference sets. As introduced in Section 4.2.2.1, for a given hypergraph, the hypergraph Laplacian L_{sym} is computed. After solving the eigenvectors of L_{sym}, a new representation of the original vertices are formed. As illustrated in Figure 4.6, after forming a matrix using the eigenvectors as columns, the rows of the matrix are taken as the new vector representations of the vertices. The vertices in the new spectral space can easily be partitioned, because they are well-separated by then.

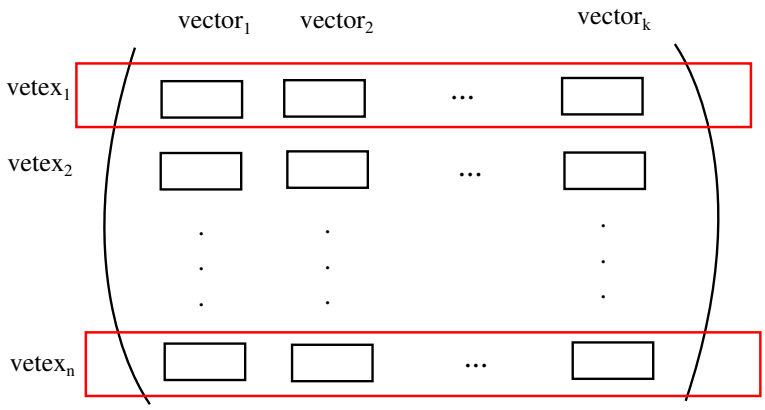

Figure 4.6: Illustration of the Spectral Embedding

The ***Hypergraph Resolver*** (i.e. the partitioner) aims to detect the intrinsic cluster structure in the hypergraph. It partitions every hypergraph into several sub-hypergraphs, each corresponding to one set of coreferent mentions (see e.g. the output in Figure 4.2(1b) which contains two sub-hypergraphs). Section 4.3.3.1 and 4.3.3.2 describe our proposed partitioning algorithms which form the core parts of the hypergraph resolver.

4.3.3.1 Recursive 2-way Partitioner

We propose the recursive variant of spectral clustering, *recursive 2-way partitioning (R2 partitioner)* (Cai & Strube, 2010a). This method does not need any information about the number of target sets (the number k of clusters). Instead a stopping criterion $\alpha^\star$ has to be provided which is adjusted on development data. At each recursion step, the *R2 partitioner* bi-partitions the focusing graph and the resulting partitions will be kept only if the cut value is smaller than $\alpha^\star$. The graph Laplacian is re-computed at each recursion based on the input graph. The algorithmic details are referred to Algorithm 1.

In the *R2 partitioner*, only one eigenvector V_2 is used for the spectral embedding and consequently the new vertex representation is only in one dimension. Therefore, directly searching for a best splitting point in V_2 is sufficient to partition the graph, with vertices ordered according to their corresponding V_2 values. For recursion purpose, all the sub-hypergraphs that can be partitioned with a *NCut* value smaller than the $\alpha*$ are partitioned further. When the *NCut* value is bigger than the $\alpha*$, it is suggesting a strong connectivity within the hypergraph in focus so that it should not be partitioned any more.

Algorithm 1 *R2 partitioner*

Note: { $L_{sym} = I - {D_v}^{-\frac{1}{2}} H W {D_e}^{-1} H^T {D_v}^{-\frac{1}{2}}$ }
Note: { $NCut(S) := vol\partial S(\frac{1}{volS} + \frac{1}{volS^c})$}
input: target hypergraph HG, predefined $\alpha^\star$
Given a HG, construct its D_v, H, W and D_e
Compute L for HG
Solve the L for the second smallest eigenvector V_2
for each splitting point in V_2 **do**
    calculate $NCut_i$
end for
Choose the splitting point with $\min_i(NCut_i)$
Generate two sub HG's
if $\min_i(NCut_i) < \alpha^*$ **then**
    for each sub HG **do**
        Bi-partition the sub HG with *R2 partitioner*
    end for
else
    Output the current sub HG
end if
output: partitioned HG

Since the mention detectors usually aim at high recall, there are a lot of system mentions which do not match with true mentions. Including system mentions into graphs results in loosely connected outliers, and *COPA* is expected to split them out as singleton clusters. Using Normalized Cut does not generate singleton clusters, hence a heuristic singleton detection strategy is proposed in *COPA*. We apply a threshold β to each node in the graph. If a node's degree is below the threshold, the node will be removed.

4.3.3.2 Flat k-way Partitioner

The *R2 partitioner* generates an optimized bi-partitioning at each recursion step. Due to its hierarchical nature, however, it is not guaranteed that the final output clusters are also globally optimized, and it does not have any intrinsic means to include global constraints to globally guide the clustering. In order to overcome these problems, we propose a flat variant of partitioner, *flatK partitioner* (see Algorithm 2). k clusters will be output simultaneously by making use of the k smallest eigenvectors of the hypergraph Laplacian L_{sym} (as in Figure 4.6).

Algorithm 2 *flatK partitioner*

1: Note: $\{ L_{sym} = I - {D_v}^{-\frac{1}{2}} H W {D_e}^{-1} H^T {D_v}^{-\frac{1}{2}} \}$
2: Note: $\{ NCut(P_k) = \sum_{1 \leq i \leq k} \frac{vol\partial V_i}{volV_i} \}$
3: **input**: target hypergraph HG, number of clusters k
4: Given a HG, construct its D_v, H, W and D_e
5: Compute L_{sym} for the HG
6: Solve the L_{sym} for the k smallest eigenvectors $v_1, ..., v_k$
7: Construct the spectral embedding $X = (v_1, \cdots, v_k)$
8: Apply k-means to the points $(x_i)_{i=1,...,n}$ to produce k clusters $C_1, ..., C_k$
9: **output**: partitioned HG with clusters $C_1, ..., C_k$

To assist the *flatK partitioner* we propose a preference-based k *model* to predict the number of entities within documents. The details of the k *model* is introduced in Section 4.3.4.

4.3.4 *k Model*: Predicting the Number of Entities

Most clustering methods for multi-cluster tasks assume the number of clusters k to be known beforehand. However, if k is not known, choosing it turns out to be a general problem for clustering algorithms, especially when partitioning noisy data. Several methods to estimate k have been proposed (for an overview see (Milligan & Cooper, 1985) and (von Luxburg, 2010)) which focus on detecting the intrinsic cluster structures from the data where clustering is viewed as an unsupervised task.

The methods of analyzing the cluster structures, such as the gap statistic (Tibshirani et al., 2001) and the stability measurements (Ben-David et al., 2006), require relatively big graphs to support valid statistics. For instance, when there are less than 100 vertices in a graph to be partitioned, the analysis methods are not able to work stably. Since documents vary largely in numbers of mentions, *COPA* seeks methods that are **not sensitive to the graph sizes** when predicting the number of entities.

In this thesis, we propose a supervised k *model* to decide on a k — the number of entities — for each hypergraph. The objective of our k *model* is to find the best k that **optimizes the end coreference performance**. The best k does not necessarily correspond to the number of true entities (the true k), when spurious system mentions are included in the hypergraphs. We address the k predicting problem with preference modeling, where two partitionings of two different k compete with each other and the better partitioning is expected to generate a better coreference performance (e.g. the F-score number). By applying the **preference modeling**, the differences between partitionings can be captured, which are less sensitive to noise than the methods solely analyzing the graph structures. In order to avoid confusion, the terms

Partitioning, *Partition* and *Cluster* are clarified via the following example.

- **mentions**
  - m_1, m_2, m_3, m_4, m_5
- a **partitioning** P_2 ($k = 2$)
  - $\{\mathrm{m}_1, \mathrm{m}_2\}$, $\{\mathrm{m}_3, \mathrm{m}_4, \mathrm{m}_5\}$
- a **partitioning** P_3 ($k = 3$)
  - $\{\mathrm{m}_1, \mathrm{m}_2\}$, $\{\mathrm{m}_3, \mathrm{m}_4\}$, $\{\mathrm{m}_5\}$
- an example **cluster**|**partition**
  - $\{\mathrm{m}_1, \mathrm{m}_2\}$

Our proposed k *model* is outlined in Algorithm 3. Given a set of possible k's for a hypergraph, a preference model is trained to find the best k with respect to the application F-score. The details of the model are described in the following subsections.

Algorithm 3 k *model* outline

Training:
Construct hypergraphs for the documents
for each hypergraph **do**
  Estimate the k range, $[k_1, k_x]$
  Decide on OneCluster
  for $k_i \in [k_2, k_x]$ **do**
    Generate a partition P_k
  end for
  Find the best partition P_{best}
  Pair the $\{P_{k_{best}}, P_{k_i}\}$, $k_{best} < k_i$, as positive training instances
  Pair the $\{P_{k_i}, P_{k_{best}}\}$, $k_i < k_{best}$, as negative training instances
end for
Build k ***model*** **from training instances**
Testing:
Construct hypergraphs for the documents
for each hypergraph **do**
  Estimate the k range, $[k_1, k_x]$
  Decide on OneCluster
  for $k_i \in [k_2, k_x]$ **do**
    Generate a partition P_k
  end for
  Pair each $\{P_{k_i}, P_{k_j}\}$, $k_i < k_j$, as testing instances
  Use the learned k *model* to annotate the instances
  Choose the best P_k using the round-robin scheme
  Output P_k
end for

Training. Before the training, a range of possible k's for each hypergraph is estimated based on the string properties of the mentions. The lower bound is set to be 1, while the upper bound is the number of different mention strings. Determining the possible k's can also be approached by including more linguistic knowledge, for instance, to set the lower bound as the number of different proper names, which are most likely to be different entities.

Since determining if a graph should be partitioned at all (as a binary decision) is easier than deciding on the best partition (as a preference decision), the cluster with $k = 1$ denoted as *OneCluster* is decided separately by simply looking at the the second cluster with $k = 2$, as opposed to the other situations in which both partitionings need to be considered. A graph with

the second cluster which generates a high *NCut* value (greater than 0.1 in our experiments) will prefer the OneCluster, and all the others will be passed to the preference model.

We partition each hypergraph built from the training data with a set of possible k's. The resulting partitioning with k_i is denoted as P_{k_i}. The k *model* aims to find the $\arg\max_{k_i} F(P_{k_i})$, where the $F(P_{k_i})$ denotes the coreference F-score when the partitioning P_{k_i} is taken.

Two partitionings are paired as one training instance, $\{P_{k_i}, P_{k_j}\}$ with $k_i < k_j$. An instance is labeled positive when $F(P_{k_i}) > F(P_{k_j})$, and negative otherwise. This way, the k *model* casts the original problem of picking the best k into a binary classification task where the preference among each pair of k's is learned.

Testing. For testing data, all pairs of partitionings $\{P_{k_i}, P_{k_j}\}$ with $k_i < k_j$ are selected as instances. The learned k *model* assigns each instance a label of positive or negative, with positive indicating the preference for P_{k_i} and negative for P_{k_j}.

To find the top k from the pairwise preference decisions, a round-robin strategy is adopted. We assign each partition P_{k_i} a confidence value $conf(P_{k_i}) = pos(P_{k_i}) - neg(P_{k_i})$, where $pos(P_{k_i})$ is how many times P_{k_i} is preferred, and $neg(P_{k_i})$ denotes the times not preferred. The top k then is simply the one with the highest confidence value.

k Model Features. There are currently only a few features used for the k model proposed in this section. For an instance $\{P_i, P_j\}$, there are features:

(1) $MaxNCut_1$: the biggest *NCut* value of partitioning P_i;

(2) $MaxNCut_2$: the biggest *NCut* value of partitioning P_j;

(3) $MaxNCutDiff$: the difference between biggest *NCut* values of the partitioning P_j and partitioning P_j;

(4) $kDiff$: the difference between the k values used for both partitioning P_i and partitioning P_j;

(5) $ConNumDiff$: the difference between the numbers of constraints violated in partitioning P_i and partitioning P_j, and the constraints used are simply the negative features used in *COPA* (see Section 5.2).

For the *k model* learner, a decision tree classifier (*J48* provided by (Witten & Frank, 2005)) is used.

4.3.5 Complexity of *COPA*

In *COPA*, the hyperedge weights are assigned using simple descriptive statistics, so that the time the *Hypergarph Resolver* needs for building the hypergraph model, transforming the hypergraph to matrices and computing the graph Laplacian matrix is not substantial. For eigensolving, we use an open source library provided by the Colt project[3]which implements a Householder-QL algorithm to solve the eigenvalue decomposition. When applied to the symmetric graph Laplacian, the complexity of the eigensolving is given by $O(n^3)$, where n is the number of the mentions in the hypergraph.

For the *R2 partitioner*, only the top two eigenvectors are required at each recursion, the decomposition can be easily improved by Lonczos algorithm which gives $O(nm)$ as the computational cost with m as the number of an equivalent (different) graph of the hypergraph. The equivalent graph here is depicted by the hypergraph Laplacian implicitly.

To sum up, the worst computational complexity of our resolving procedure gives $O(n^3)$ and in hierarchical manner it is only $O(nm)$. Spectral clustering only becomes problematic when the graph has millions of vertices. However, for documents where at most hundreds of mentions appear it is not an issue at all.

4.4 Implementation Issues

4.4.1 The Post-processing For Pronoun Anaphors

In a hypergraph built by *COPA*, pronouns are connected to all other non-pronouns which do not violate any agreement relations, such as gender and number agreements. In an end-to-end setting, there are many singleton entities included into the hypergraphs via their connections to pronouns. As mentioned before, a spectral clustering algorithm is unable to separate singletons during partitioning, thus we may derive clusters mixed with singleton entities. In order to address this issue, we propose a post-processing strategy. For a pronoun anaphor, only its strongest connection within its assigned cluster is kept and all other links are removed.

Figure 4.7 gives an example for the post-processing of pronouns, the graph is shown in a standard graph form for the sake of clarity. The dashed (red) circles indicate the cluster boundaries.

[3]`http://acs.lbl.gov/~hoschek/colt/`

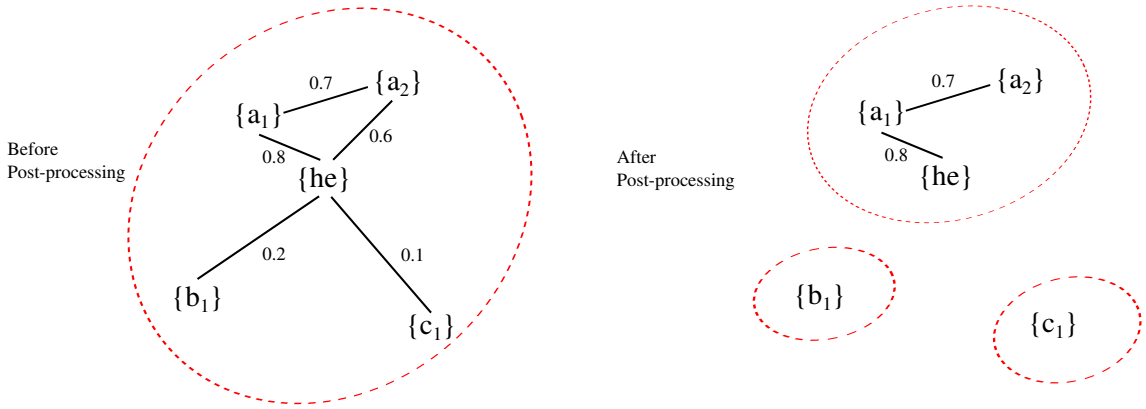

Figure 4.7: Illustration of the Post-processing for Pronouns

Considering the generated cluster in the left side of Figure 4.7 which contains the mentions $\{a_1\}$, $\{a_2\}$, $\{he\}$, $\{b_1\}$, $\{c_1\}$, with links between $\{he\}$ and all the other mentions and one link between $\{a_1\}$ and $\{a_2\}$. Assuming the strongest connection to $\{he\}$ is $\{a_1\}$, the proposed post-processing removes $\{b_1\}$ and $\{c_1\}$ while leaving $\{a_1\}$, $\{a_2\}$, $\{he\}$ in the final cluster. This post-processing is driven by the intuition that the connections between pronouns and non-pronouns are not confident enough to support transitive closures. For instance, the links between $\{he\}$ and $\{b_1\}$, $\{c_1\}$ are not confident enough to enforce a connection between $\{b_1\}$ and $\{c_1\}$. We only maintain one link per pronoun after the partitioning procedure, e.g. the one between $\{he\}$ and $\{a_1\}$, but keeping other relations being transitive so that $\{a_2\}$ is also in the final cluster.

4.4.2 Partitioning Issues

Graph Components. The number of zero eigenvalues corresponds to the number of components in the graph (von Luxburg, 2007). A graph component is a disconnected sub-graph, and in *COPA* multiple components can occur when only limited features are used, so that not all mentions from the document are connected (directly or via a path). Different components can be processed separately during partitioning process, for the sake of reducing complexity. Only for the connected graphs, the top (k) eigenvectors are taken as described for the spectral embedding.

Eigenvalue Smoothing. It is worth noting that depending on the implementation details of the eigen decomposition component, the solved eigenvalues can be a double or a float type. It is necessary to smooth the eigenvalues, for instance by applying an Epsilon variable (e.g. a small number) to allow for small fluctuations on the eigenvalues.

The k-means Initialization. It is well known that the k-means algorithm is sensitive to the initialization of cluster centers. Since there is a lot of noise involved in our hypergraphs, the decision on the initial cluster centers becomes even more crucial. Accidentally choosing the noisy mentions as initial centers can generate unexpected clusters. In *COPA*, we address this issue by restricting the initial cluster centers to proper names that are more likely to lead entities. This modification manages to introduce application specific knowledge into the k-means to guide the initialization, and can be easily improved by estimating the entity centers using more information.

4.5 Hypergraphs to Standard Graphs

The hypergraph is a generalization of the standard graph. It is possible to find graphs which approximate hypergraphs and thus can be accessed using the standard graph-based algorithms. In order to preserve the power of representation of the hypergraph, in *COPA* we avoid the transformation step by applying the partitioning algorithm directly to the hypergraph models. However, in this section, we introduce the equivalent graphs to the hypergraph, which serve as **alternatives** when hypergraph-based algorithms are not available or when one wants to explore more inference models upon the hypergraph representation.

The two most commonly used ones are *Star Expansion* and *Clique Expansion* (Agarwal et al., 2005). *Star Expansion* (in Section 4.5.1) introduces a new star vertex for each hyperedge, which connects all the vertices covered by the original hyperedge. As a result, a bi-partite graph is generated where the edge weights can be assigned by distributing the corresponding hyperedge weights evenly. *Clique Expansion* (in Section 4.5.2) expands each hyperedge into cliques, and the similarity between two vertices is proportional to the summed weights of their common labels.

4.5.1 The Star Expansion

Star Expansion transforms the hypergraph into a bi-partite graph, where there are additional starred vertices corresponding to original hyperedges. All the vertices belonging to a hyperedge are therefore connected to the new starred vertex in the bi-partite graph. The weights of the multiple edges generated from one hyperedge e is normalized by the degree of e:

$$w'(u, e) = w(e)/\delta(e) \tag{4.11}$$

where the $w(e)$ is the original hyperedge weight and u is a vertex connecting to e.

4.5.2 The Clique Expansion

Clique Expansion transforms each hyperedge into several pairwise edges (Zien et al., 1999), so that the vertices in a hyperedge form a clique. The new edge weights between vertex u and v is

$$w'(u,v) = \mu \sum_{e} h(u,e)h(v,e)w(e) \tag{4.12}$$

where the $w(e)$ is the original hyperedge weight and μ is a fixed scalar.

4.6 Summary

Our Contributions. In this chapter, we introduce our proposed coreference resolution model — *COPA*, standing for coreference partitioner. Our contributions are two-fold, (1) representing the coreference relation with the **hypergraph model**, and (2) inferring coreference sets using the **hypergraph partitioning** algorithms.

COPA represents documents in the hypergraph model, so that the multiple low-dimensional relations between mentions are easily expressed as hyperedges without the necessity of combining them before the final decision. Upon the constructed hypergraphs, the spectral clustering technique is applied to derive coreference sets directly and simultaneously. By adopting spectral clustering algorithms, it is made sure that the mentions within a coreference set are closely related, while the ones from different sets are far apart from each other.

Spectral Hypergraph Partitioning for Coreference Resolution. The proposed hypergraph partitioning model looks at the entire graph to make coreference decisions. Not only the context preceding a mention but also the one after it are evaluated to assign the mention to one of the clusters. We propose two partitioning algorithms for *COPA*, the ***R2 partitioner*** performs the hierarchical clustering and the ***flatK partitioner*** partitions only once. To assist the *flatK partitioner*, we propose a novel k ***model*** to predict the number of entities within documents.

End-to-end Coreference Resolution. We address the coreference resolution problem in an end-to-end system setup, where noise is unavoidable and the mentions to be resolved may not align with the true mention set. Implementing coreference models in end-to-end systems is very important, since it has been observed that improved performance on true mentions does not necessarily translate into the improved performance on system mentions (Ng, 2008). The **implementation issues** of applying clustering techniques to coreference resolution are addressed in this chapter too.

Overall, the hypergraph representation of *COPA* avoids the expensive training for the feature combination, and its light weighted partitioning-based inference does not ask for complex probabilistic estimations. *COPA*'s partitioning-based strategy can be taken as a general preference model, where the preference of entities for one mention depends on information on all other mentions. Therefore, we believe that *COPA* is a coreference model preferable not only to the previous local models but also to complicated graphical methods.

Chapter 5

COPA Features

In this chapter, we introduce the feature representation scheme encoded in *COPA*. Our features aim to capture the linguistic phenomena of the coreference relation, as well as the data-specific statistics. *COPA* has been applied to various types of data sets ranging from news articles (e.g. MUC, ACE and OntoNotes data sets in Chapter 3) to clinical reports (e.g. the I2B2 corpus), the feature sets it implements therefore cover both general and domain-specific information.

5.1 The Feature Categorization in the Hypergraph

Positive relational features can be incorporated into the hypergraph model of *COPA* as types of hyperedges (e.g. in Figure 4.2 (b) the two hyperedges marked by "– ··" are of the same type from feature *subject/object match*), so that a realized hyperedge is an instance of a corresponding type. All hyperedge instances that are derived from the same type have the same weight, but they may get re-weighted by the distance feature (Section 5.5). Negative relations can be treated either as filters to be applied to the graph construction phase (e.g. the negative features described in Section 5.2) or as constraints to be applied to the inference procedure (see Chapter 8). In this chapter, we only focus on the features adopted for constructing the hypergraphs, which consist of three categories:

Negative Features: to prevent hyperedges between mentions;

Positive Features: to generate relatively strong hyperedges between mentions;

Weak Features: to add hyperedges to an existing hypergraph without introducing new mentions into the hypergraph;

Negative features here act as global filtering variables, avoiding incompatible mentions to be connected in a graph. For instance, although [*Mr. Clinton*] and [*Mrs. Clinton*] match

via *substring match* (positive) feature, there is no hyperedge built between them due to their incompatible gender.

COPA differentiates between positive and weak features, because spectral clustering algorithms do not have intrinsic means to handle singleton clusters. Recall that the spectral clustering technique targets at optimizing the normalized cut (NCut) value, which has the inner-cluster connectives factor as the denominator. This therefore makes it impossible to output singleton clusters. In order to avoid too much noise (e.g. singleton mentions) in our hypergraph model, we construct the graphs in a conservative manner. While weak relations contribute to the graph structure, they tend to involve too many singleton mentions into the graph. So we construct hypergraphs solely out of the positive features and only add weak relations into the graph afterward without introducing new vertices at all.

In the following sections we describe the features implemented in *COPA*.

5.2 Negative Features

Negative features describe the pairwise relations between mentions that are most likely to be not coreferent. They have been conventionally used in combination with other features (Soon et al., 2001) and is implemented as weak positive features in an early version of *COPA* (Cai & Strube, 2010a). Now we apply negative features as global filters in the graph construction phase. When mentions are detected to be in a negative relation, it is made sure that no edges are built between them in the hypergraphs.

(1) N_Gender, (2) N_Number

Two mentions do not agree in gender or number.

For instance, no edge is allowed between the mentions [*Hillary Clinton*] and [*he*] due to their incompatible gender. The mention [*Mr. Sisulu*] has the negative relation of incompatible number with the mention [*boys*].

(3) N_SemanticClass

Two mentions do not agree in semantic classe.

For news articles (e.g. MUC, ACE and OntoNotes data sets), only the *Object*, *Date*, *Person* and other top categories derived from WordNet (Fellbaum, 1998) are used. For clinical reports (e.g. I2B2 corpus), this feature is replaced by feature (7) that identifies the medical types for each mention.

(4) N_Mod

Two mentions have the same syntactic heads, and the anaphor has a modifier that does not occur in the antecedent or contradicts the modifiers of the antecedent.
For instance, a negative relation is built between the mentions [*expedited proceedings*] and [*the investigation proceedings*], as the modifiers of the two mentions convey different information. However, simply enforcing the modifiers to be the same cannot handle the situations in which the modifiers appear differently though without contradicting each other (e.g. [*the case in question*] and [*the case against the accused*]). The current version of *COPA* does not take care of these difficult cases.

(5) N_DSPrn

Two first person pronouns (i.e. [*I*], [*me*], [*my*] etc.) in direct speech which are assigned to different speakers should not be linked together. The speaker information is given in the OntoNotes data set.

(6) N_ContraSubjObj

Two mentions are in the subject and object positions of the same verb, and the anaphor is not a possessive pronoun.
For instance, [*John*] talks to [*him*], where [*John*] should not be coreferent with [*him*].

(7) N_i2b2Type

Two mentions have different mention types (e.g. *treatment*, *problem*, etc. as defined in the I2B2 data set).
For instance, [*Ischemic bowel*] has an incompatible I2B2 type with [*Thoracentesis*], as a clinical problem mention cannot be coreferent with a medical treatment mention.

(8) N_i2b2Quant

Two mentions are modified by different quantities.
For instance, the mention [*heart rate*] in the text fragment ”heart rate 116” and the mention [*a heart rate*] in the text fragment ”a heart rate of 128” cannot be coreferent.

(9) N_i2b2ConName

Two mentions have the same syntactic heads, and their matched (if ever) concept names in MetaMap are different.
For example, the mention [*back pain*] and the mention [*chest pain*] are in this negative relation.

5.3 Positive Features

The majority of the well-studied coreference features (e.g. Stoyanov et al. (2009)) are positive coreference indicators. In our system, the mentions that participate in positive relations are included in the hypergraphs as vertices.

(10) StrMatch_Npron & (11) StrMatch_Pron

After discarding stop words, if the strings of mentions completely match and are not pronouns, they are put into hyperedges of the *StrMatch_Npron* type. When the matched mentions are pronouns, they are connected with a *StrMatch_Pron* hyperedge. We differentiate the two types of string matchings, as pronouns suggest much less information than non-pronouns do.

(12) Alias

After discarding stop words, if mentions are aliases of each other (i.e. proper names with partial match, full names and acronyms of organizations, etc.).
For instance, [*Australia's Qintex*] and [*Qintex Australia Ltd.*] are aliases of each other.

(13) HeadMatch

If the syntactic heads of mentions match, such as [*the U.S. rules*] and [*the rules*].

(14) Nprn_Prn

If the antecedent is not a pronoun and the anaphor is a pronoun. The feature is designed with the intuition that pronouns are used to refer to existing entities. Although this feature is not highly weighted, it is crucial for integrating pronouns into the hypergraph.

(15) Speaker12Prn

If the speaker of a second person pronoun is talking to the speaker of a first person pronoun, the two pronouns are connected with a hyperedge. This type of hyperedges only contain first and second person pronouns. This feature is useful for the OntoNotes data set where speaker information (e.g. the speaker names and the speech boundaries) is explicitly provided.

(16) DSPrn

If one of the mentions is the subject of a *speak* verb, and other mentions are first person pronouns within the corresponding direct speech. Direct speech boundaries are detected simply by paring double quotes.

(17) ReflexivePrn
If the anaphor is a reflexive pronoun, and the antecedent is the subject of the same clause. Dependency trees are utilized to conduct the necessary grammatical analysis.
In sentence ”[*today’s generation of Taiwanese*] save our island’s last remaining forest of these giant trees, for [*themselves*] and later generations?”, the marked mentions are linked via this feature.

(18) PossPrn
If the anaphor is a possessive pronoun, and the antecedent is the subject in the same sub-clause.
In sentence ”How would you feel if [*your child*] learned from [*his*] classmates to cough up phlegm all over the place?”, the marked mentions are in this relation.

(19) GPEIsA
If the antecedent is a Named Entity of GPE entity types (i.e. one of the ACE entity type (NIST, 2004)), and the anaphor is a definite expression of the same type.
For instance, [*Iraq*] is linked with [*the nation*].

(20) OrgIsA If the antecedent is a Named Entity of Organization entity type, and the anaphor is a definite expression of the same type.
For instance, [*Google Inc.*] is linked with [*the company*].

Feature (19) and (20) capture the IsA relations for specific types of Named Entities, and are designed for news article data sets.

(21) Appositive
Two mentions are in an appositive structure, such as the mention [*Laurence Tribe, Gore’s attorney*] and its embedded mention [*Gore’s attorney*]. Depending on the annotation schemes of the adopted data set, this relation may or may not be a coreference indicator.

(22) Concept
We disambiguate each Named Entity to Wikipedia entries (Fahrni et al., 2012), and if mentions linked to the same entries.
For instance, [*South Korea*] and [*ROK*] are disambiguated to the same entry so that they are connected by this feature.

(23) i2b2PisA

A pseudo IsA relation. One mention appears in other mentions' definitions extracted from the UMLS thesaurus.
For instance, the mentions [*Paracentesis*] and [*the tap*] are captured by this feature, since the top ranked definition of [*the tap*] is "Paracentesis".

(24) i2b2Abbr

One mention is in the abbreviation format (i.e. with all letters capitalized), the other mentions match (exactly or partially) with its concept name extracted by MetaMap.
For instance, the mention [*EGD*] is identified to be the abbreviation of the mention [*esophagogastroduodenoscopy*].

(25) i2b2CatMatch

There is always structured information in the clinical data sets (e.g. I2B2), as shown in the text "[*Attending*]: [*Erm K. Neidwierst , M.D.*]". The mentions are linked when they appear in the same category slot of the report and both are persons.

(26) i2b2PrnPreference

This is a data specific feature, describing the preferences for certain types of pronouns.
For example, first person singular pronouns in the data set mostly refer to the physician who writes the clinical report.

5.4 Weak Features

Weak features are weak coreference indicators. Using them as positive features would introduce too much noise to the graph (i.e. a graph with too many singletons). We apply weak features only to mentions already integrated in the graph, so that weak information provides it with a richer structure.

(27) W_VerbAgree

If the anaphor is a pronoun, and the antecedent appears as a subject or an object in previous sentences. The verbs of both mentions should be the same.
For instance, the sentence "Born in Homei, Changhua in 1928, [*Hsu*] studied the violin in Japan as a youth" is followed by the sentence "Later, [*he*] studied in France ...", so that the marked two mentions share this *W_VerbAgree* relation.

(28) W_Subject

If mentions are subjects.

(29) W_Synonym
If mentions are synonymous as indicated by WordNet, such as [*the town*] and [*the village*].

(30) W_i2b2SubStr
One mention is the substring of the other.
For instance, the mention [*Cisplatin*] is the substring of the mention [*Cisplatin chemotherapy*].

5.5 The Distance Feature

Graph models cannot deal with positional information well, such as distance between mentions or the sequential ordering of mentions in a document. Therefore the hypergraph model of *COPA* does not have any obvious means to encode distance information. However, distance between mentions plays an important role in coreference resolution, especially for resolving pronouns. We do not encode distance as a binary feature, as this introduces too many hyperedges into the graph. Instead, we use distance to re-weigh hyperedges of degrees of 2, which are supposed to be sensitive to positional information.

We experiment with two types of distance weights: **(31) sentence distance** as used in Soon et al. (2001)'s feature set and **(32) compatible mentions distance** as introduced by Bengtson & Roth (2008).

5.6 The Learned Hyperedge Weights

Table 5.1 and Table 5.2 provide the example feature weights (i.e. hyperedge weights) learned from the OntoNotes training set, in order to indicate the hypergraph structures we derived. I2B2-relevant feature weights are shown in Table 5.3. In Table 5.4, the statistics for the negative features suggest how strongly the features are contributing to non-coreference decisions.

OntoNotes data does not annotate appositive relations as coreference relations, so that Feature (21) gives surprisingly small weights.

Positive Features	Weights
(10) StrMatch_Npron	0.766
(11) StrMatch_Pron	0.620
(12) Alias	0.733
(13) HeadMatch	0.614
(14) Nprn_Prn	0.176
(15) Speaker12Prn	0.552
(16) DSPrn	0.9
(17) ReflexivePrn	0.567
(18) PossPrn	0.75
(19) GPEIsA	0.308
(20) OrgIsA	0.111
(21) Appositive	0.001
(22) Concept	0.494

Table 5.1: Positive Feature Weights on OntoNotes Data

Weak Features	Weights
(27) W_VerbAgree	0.342
(28) W_Subject	0.4425
(29) W_Synonym	0.429

Table 5.2: Weak Feature Weights on OntoNotes Data

I2B2 Features	Weights
(23) i2b2PisA	0.348
(24) i2b2Abbr	0.423
(25) i2b2CatMatch	0.935
(26) i2b2PrnPreference	0.967
(30) W_i2b2SubStr	0.594

Table 5.3: Feature Weights on I2B2 Data

Negative Features	Statistics
(1) N_Gender	-0.993
(2) N_Number	-0.996
(3) N_SemanticClass	-0.993
(4) N_Mod	-0.853
(5) N_DSPrn	-0.762
(6) N_ContraSubjObj	-0.997
(7) N_i2b2Type	-0.999
(8) N_i2b2Quant	-0.999
(9) N_i2b2ConName	-0.816

Table 5.4: Negative Feature Statistics on OntoNotes Data

5.7 Summary

In *COPA*, features are expressed as hyperedges. Since the combination of features is implicitly done during the inference phase, the features in the graph construction phase simply are included in an overlapping manner. Therefore it is straightforward and costs little effort to include more features in *COPA*. We categorize the features into three types, which do not only indicate the linguistic functions of different features but also provide a systematic way for feature development in *COPA*.

Negative relations are interpreted as global filters during the graph construction in this Chapter, and they are explored further in Chapter 8 as global constraints which are applied during the inference phase. Coreference decisions depend on preferences, where negative information in certain cases contributes as much as the conventional positive indicators.

Chapter 6

Evaluation Metrics for End-to-end Coreference Resolution

Evaluating clustering results is one of the most important issues in cluster analysis, and is referred as clustering validation (Halkidi et al., 2001). When the ground truth is provided, the evaluation methods aim to measure how similar the clustering results are to the gold annotations. For instance, the evaluation metrics for coreference resolution measure the output coreference sets (i.e. clusters) against the ground truth sets provided by domain experts. Since there may be different numbers of output clusters (e.g. the coreference sets) compared with the gold annotations, such an evaluation task is different from evaluating classification problems which directly assesses the label assignments of instances. It becomes more complicated to perform the evaluation when the numbers of the output instances (e.g. the mentions) are also different from the gold ones. In this chapter, we focus on the end-to-end system setting for the coreference resolution task, and propose evaluation algorithms to assess noisy coreference output.

Early research on coreference resolution has worked on the *true mention* setting, where the mentions participating in coreference sets are given along with their exact boundaries. The commonly used coreference resolution evaluation metrics are designed for such systems, but evaluate the output coreference sets from different perspectives. For instance, the MUC score (Vilain et al., 1995) in Section 6.1.1 performs on the relations between mentions, the B^3 algorithm (Bagga & Baldwin, 1998) in Section 6.1.2 operates on the relations between mentions and sets, and the *CEAF* algorithm (Luo, 2005) in Section 6.1.3 captures the relations between sets. However, it is not trivial to apply these metrics to end-to-end coreference systems, where the automatically identified *system mentions* may not align with the true mentions. To be consistent with the literature, in this chapter *key mention* is used to refer to true mention.

In Section 6.1, we discuss the problems of the existing coreference metrics and propose

two variants of the B^3 and *CEAF* algorithms which can be applied to noisy coreference output dealing with system mentions. Our experiments in Section 6.2 show that our variants lead to intuitive and reliable results for end-to-end coreference systems.

6.1 Evaluation Metrics for the End-to-end Coreference Resolution

6.1.1 *MUC*

The MUC score (Vilain et al., 1995) counts the minimum number of links between mentions to be inserted or deleted when mapping a system response to a gold standard key set. Given an example,

Key : $\{m_1, m_2, m_3, m_4\}$

Response: $\{m_1, m_2\}$ $\{m_3, m_4\}$

Figure 6.1 illustrates the relations between mentions for both the key and the response. Since the response sets require at least one link (e.g. between m_1 and m_4) to form a set (i.e. $\{m_1, m_2, m_3, m_4\}$) which matches the provided key, the recall is given as $Recall = 2/3$. The precision is computed as $Precision = 2/2$, as all the links in the response are correct.

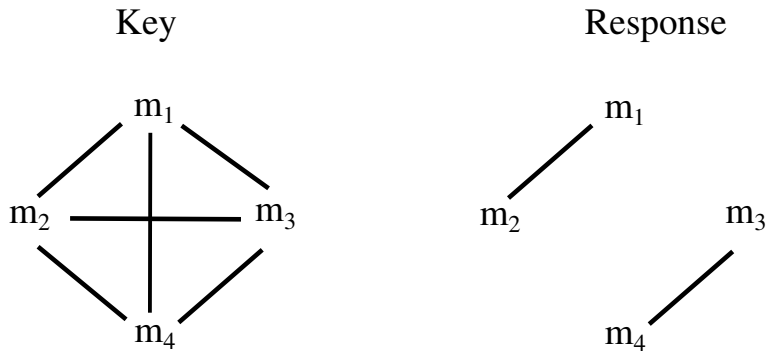

Figure 6.1: The MUC Score Illustration

Although pairwise links capture the relations in a set, they cannot represent singleton entities, i.e. entities, which are mentioned only once. Therefore, the MUC score is not suitable for the ACE data (`http://www.itl.nist.gov/iad/mig/tests/ace/`), which includes singleton entities in the keys. Moreover, the MUC score does not give credit for separating singleton entities from other chains. This becomes problematic in a realistic system setup, when mentions are extracted automatically.

6.1.2 B^3

The B^3 algorithm (Bagga & Baldwin, 1998) overcomes the shortcomings of the MUC score. Instead of looking at the links, B^3 computes precision and recall for all mentions in the document, which are then combined to produce the final precision and recall numbers for the entire output. For each mention, the B^3 algorithm computes a precision and recall score using equations 6.1 and 6.2:

$$Precision(m_i) = \frac{|R_{m_i} \cap K_{m_i}|}{|R_{m_i}|} \tag{6.1}$$

$$Recall(m_i) = \frac{|R_{m_i} \cap K_{m_i}|}{|K_{m_i}|} \tag{6.2}$$

where R_{m_i} is the response chain (i.e. the system output) which includes the mention m_i, and K_{m_i} is the key chain (manually annotated gold standard) with m_i. The overall precision and recall are computed by averaging them over all mentions.

Considering the same example as in the previous section,

Key : {m_1, m_2, m_3, m_4}

Response: {m_1, m_2} {m_3, m_4}

Figure 6.2 illustrates the relations between mentions and their corresponding sets.

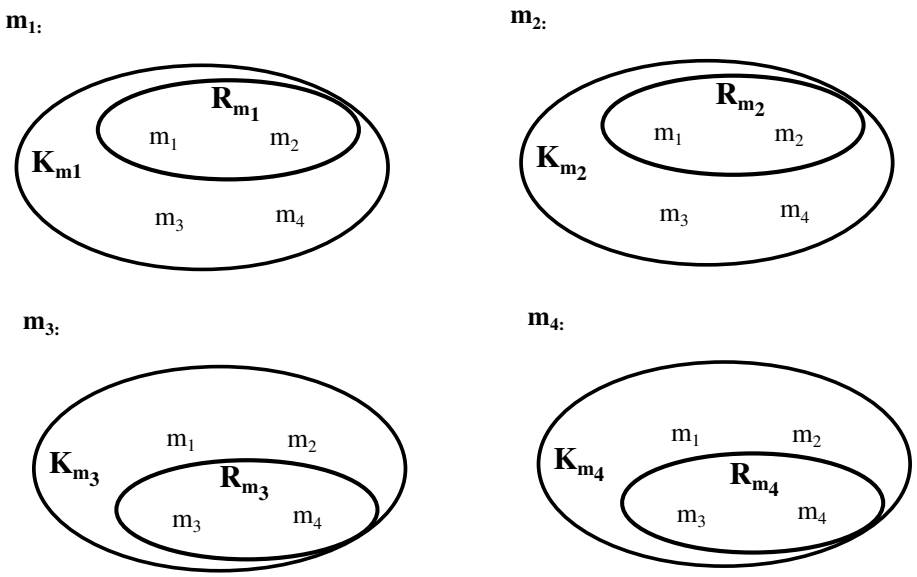

Figure 6.2: The B^3 Algorithm Illustration

According to Equation 6.1 and 6.2,

$$Precision(m_1) = \frac{2}{2}, Recall(m_1) = \frac{2}{4}$$

$$Precision(m_2) = \frac{2}{2}, Recall(m_2) = \frac{2}{4}$$

$$Precision(m_3) = \frac{2}{2}, Recall(m_3) = \frac{2}{4}$$

$$Precision(m_4) = \frac{2}{2}, Recall(m_4) = \frac{2}{4}$$

Since B^3's calculations are based on mentions, singletons are taken into account. However, a problematic issue arises when system mentions have to be dealt with: B^3 assumes the mentions in the key and in the response to be identical. Hence, B^3 has to be extended to deal with system mentions which are not in the key and key mentions not extracted by the system, so called *twinless mentions* (Stoyanov et al., 2009).

6.1.2.1 Existing B^3 variants

A few variants of the B^3 algorithm for dealing with system mentions have been introduced recently. (Stoyanov et al., 2009) suggest two variants of the B^3 algorithm to deal with system mentions, B^3_0 and B^3_{all}[1]. For example, a key and a response are provided as below:

Key : {a b c}

Response: {a b d}

B^3_0 discards all twinless system mentions (i.e. mention d) and penalizes recall by setting $recall_{m_i} = 0$ for all twinless key mentions (i.e. mention c). The B^3_0 precision, recall and F-score (i.e. $F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$) for the example are calculated as:

$$Pr_{B^3_0} = \frac{1}{2}(\frac{2}{2} + \frac{2}{2}) = 1.0$$

$$Rec_{B^3_0} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + 0) \doteq 0.444$$

$$F_{B^3_0} = 2 \times \frac{1.0 \times 0.444}{1.0 + 0.444} \doteq 0.615$$

B^3_{all} retains twinless system mentions. It assigns $1/|R_{m_i}|$ to a twinless system mention as its precision and similarly $1/|K_{m_i}|$ to a twinless key mention as its recall. For the same example above, the B^3_{all} precision, recall and F-score are given by:

$$Pr_{B^3_{all}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$$

[1] Our discussion of B^3_0 and B^3_{all} is based on the analysis of the source code available at `http://www.cs.utah.edu/nlp/reconcile/`.

$$Rec_{B^3_{all}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$$

$$F_{B^3_{all}} = 2 \times \frac{0.556 \times 0.556}{0.556 + 0.444} \doteq 0.556$$

Tables 6.1, 6.2 and 6.3 illustrate the problems with B^3_0 and B^3_{all}. The rows labeled *System* give the original keys and system responses while the rows labeled B^3_0, B^3_{all} and B^3_{sys} show the performance generated by Stoyanov et al.'s variants and the one we introduce in this chapter, B^3_{sys} (the row labeled $CEAF_{sys}$ is discussed in Subsection 6.1.3).

		Set 1		
System 1	key	{a b c}		
	response	{a b d}		
		P	R	F
B^3_0		1.0	0.444	0.615
B^3_{all}		0.556	0.556	0.556
$B^3_{r\&n}$		0.556	0.556	0.556
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572
System 2	key	{a b c}		
	response	{a b d e}		
		P	R	F
B^3_0		1.0	0.444	0.615
B^3_{all}		0.375	0.556	0.448
$B^3_{r\&n}$		0.375	0.556	0.448
B^3_{sys}		0.5	0.556	0.527
$CEAF_{sys}$		0.4	0.667	0.500

Table 6.1: Problems of B^3_0

In Table 6.1, there are two system outputs (i.e. *System 1* and *System 2*). Mentions *d* and *e* are the twinless system mentions erroneously resolved and *c* a twinless key mention. *System 1* is supposed to be slightly better with respect to precision, because *System 2* produces one more spurious resolution (i.e. for mention *e*). However, B^3_0 computes exactly the same numbers for both systems. Hence, there is no penalty for erroneous coreference relations in B^3_0, if the mentions do not appear in the key, e.g. putting mentions *d* or *e* in *Set 1* does not count as precision errors. — B^3_0 is too lenient by only evaluating the correctly extracted mentions.

		Set 1	Singletons	
System 1	key	{a b c}		
	response	{a b d}		
		P	R	F
B^3_{all}		0.556	0.556	0.556
$B^3_{r\&n}$		0.556	0.556	0.556
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572
System 2	key	{a b c}		
	response	{a b d}	{c}	
		P	R	F
B^3_{all}		0.667	0.556	0.606
$B^3_{r\&n}$		0.667	0.556	0.606
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572

Table 6.2: Problems of B^3_{all} (1)

		Set 1	Singletons	
System 1	key	{a b}		
	response	{a b d}		
		P	R	F
B^3_{all}		0.556	1.0	0.715
$B^3_{r\&n}$		0.556	1.0	0.715
B^3_{sys}		0.556	1.0	0.715
$CEAF_{sys}$		0.667	1.0	0.800
System 2	key	{a b}		
	response	{a b d}	{i} {j} {k}	
		P	R	F
B^3_{all}		0.778	1.0	0.875
$B^3_{r\&n}$		0.556	1.0	0.715
B^3_{sys}		0.556	1.0	0.715
$CEAF_{sys}$		0.667	1.0	0.800

Table 6.3: Problems of B^3_{all} (2)

B^3_{all} deals well with the problem illustrated in Table 6.1, the figures reported correspond to intuition. However, B^3_{all} can output different results for identical coreference resolutions when exposed to different mention taggers as shown in Tables 6.2 and 6.3. B^3_{all} manages to penalize erroneous resolutions for twinless system mentions, however, it ignores twinless key mentions when measuring precision. In Table 6.2, *System 1* and *System 2* generate the same output, except that the mention tagger in *System 2* also extracts mention *c*. Intuitively, the same numbers are expected for both systems. However, B^3_{all} gives a higher precision to *System 2*, which results in a higher F-score.

B^3_{all} retains all twinless system mentions, as can be seen in Table 6.3. *System 2*'s mention tagger tags more mentions (i.e. the mentions *i*, *j* and *k*), while both *System 1* and *System 2* have identical coreference resolution performance. Still, B^3_{all} outputs quite different results for precision and thus for F-score. This is due to the credit B^3_{all} takes from unresolved singleton twinless system mentions (i.e. mention *i*, *j*, *k* in *System 2*). Since the metric is expected to evaluate the end-to-end coreference system performance rather than the mention tagging quality, it is not satisfying to observe that B^3_{all}'s numbers actually fluctuate when the system is exposed to different mention taggers.

Rahman & Ng (2009) apply another variant, denoted here as $B^3_{r\&n}$. They remove only those twinless system mentions that are singletons before applying the B^3 algorithm. So, a system would not be rewarded by the the spurious mentions which are correctly identified as singletons during resolution (as has been the case with B^3_{all}'s higher precision for *System 2* as can be seen in Table 6.3).

We assume that Rahman & Ng apply a strategy similar to B^3_{all} after the removing step (this is not clear in Rahman & Ng (2009)). While it avoids the problem with singleton twinless system mentions, $B^3_{r\&n}$ still suffers from the problem dealing with twinless key mentions, as illustrated in Table 6.2.

6.1.2.2 Our proposed variant — B^3_{sys}

We here propose a coreference resolution evaluation metric, B^3_{sys}, which deals with system mentions more adequately (see the rows labeled B^3_{sys} in Tables 6.1, 6.2, 6.3, 6.8 and 6.9). We put all twinless key mentions into the response as singletons which enables B^3_{sys} to penalize non-resolved coreferent key mentions without penalizing non-resolved singleton key mentions, and also avoids the problem B^3_{all} and $B^3_{r\&n}$ have as shown in Table 6.2. All twinless system mentions that are deemed not coreferent (hence being singletons) are discarded. To calculate B^3_{sys} precision, all twinless system mentions that are mistakenly resolved are put into the key since they are spurious resolutions (equivalent to the assignment operations in B^3_{all}), which should be penalized by precision. Unlike B^3_{all}, B^3_{sys} does not benefit from unresolved twinless system mentions (i.e. the twinless singleton system mentions). For recall, the algo-

rithm only goes through the original key sets, similar to B^3_{all} and $B^3_{r\&n}$. Details are given in Algorithm 4.

Algorithm 4 B^3_{sys}

Input: key sets key, response sets $response$

Output: precision P, recall R and F-score F

1: Discard all the singleton twinless system mentions in $response$;
2: Put all the twinless annotated mentions into $response$;
3: **if** calculating precision **then**
4: Merge all the remaining twinless system mentions with key to form key_p;
5: Use $response$ to form $response_p$
6: Through key_p and $response_p$;
7: Calculate B^3 precision P.
8: **end if**
9: **if** calculating recall **then**
10: Discard all the remaining twinless system mentions in $response$ to from $response_r$;
11: Use key to form key_r
12: Through key_r and $response_r$;
13: Calculate B^3 recall R
14: **end if**
15: Calculate F-score F

For example, a coreference resolution system has the following key and response:

Key : {a b c}

Response: {a b d} {i j}

To calculate the precision of B^3_{sys}, the key and response are altered to:

Key$_p$: {a b c} {d} {i} {j}

Response$_p$: {a b d} {i j} {c}

So, the precision of B^3_{sys} is given by:

$$Pr_{B^3_{sys}} = \frac{1}{6}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{2} + 1) \doteq 0.611$$

The modified key and response for recall are:

Key$_r$: {a b c}

Response$_r$: {a b} {c}

The resulting recall of B^3_{sys} is:

$$Rec_{B^3_{sys}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$$

Thus the F-score number is calculated as:

$$F_{B^3_{sys}} = 2 \times \frac{0.611 \times 0.556}{0.611 + 0.556} \doteq 0.582$$

B^3_{sys} indicates more adequately the performance of end-to-end coreference resolution systems. It is not easily tricked by different mention taggers. Further example analysis for the proposed B^3_{sys} can be found in Section 6.1.2.3.

6.1.2.3 B^3_{sys} Example Output

Here, we provide additional examples for analyzing the behavior of B^3_{sys} where we systematically vary system outputs. Since we propose B^3_{sys} for dealing with end-to-end systems, we consider only examples also containing twinless mentions. The systems in Table 6.4 and 6.6 generate different twinless key mentions while keeping the twinless system mentions untouched. In Table 6.5 and 6.7, the number of twinless system mentions changes through different responses and the number of twinless key mentions is fixed.

In Table 6.4, B^3_{sys} recall goes up when more key mentions are resolved into the correct set. And the precision stays the same, because there is no change in the number of the erroneous resolutions (i.e. the spurious cluster with mentions i and j). For the examples in Tables 6.5 and 6.7, B^3_{sys} gives worse precision to the outputs with more spurious resolutions, but the same recall if the systems resolve key mentions in the same way. Since the set of key mentions intersects with the set of twinless system mentions in Table 6.6, we do not have an intuitive explanation for the decrease in precision from response$_1$ to response$_4$. However, both the F-score and the recall still show the right tendency.

	Set 1	Set 2	B^3_{sys}		
key	{a b c d e}		P	R	F
response$_1$	{a b}	{i j}	0.857	0.280	0.422
response$_2$	{a b c}	{i j}	0.857	0.440	0.581
response$_3$	{a b c d}	{i j}	0.857	0.68	0.784
response$_4$	{a b c d e}	{i j}	0.857	1.0	0.923

Table 6.4: Analysis of B^3_{sys} 1

	Set 1	Set 2	B^3_{sys}		
key	{a b c d e}		P	R	F
response$_1$	{a b c}	{i j}	0.857	0.440	0.581
response$_2$	{a b c}	{i j k}	0.75	0.440	0.555
response$_3$	{a b c}	{i j k l}	0.667	0.440	0.530
response$_4$	{a b c}	{i j k l m}	0.6	0.440	0.508

Table 6.5: Analysis of B^3_{sys} 2

	Set 1	B^3_{sys}		
key	{a b c d e}	P	R	F
response$_1$	{a b i j}	0.643	0.280	0.390
response$_2$	{a b c i j}	0.6	0.440	0.508
response$_3$	{a b c d i j}	0.571	0.68	0.621
response$_4$	{a b c d e i j}	0.551	1.0	0.711

Table 6.6: Analysis of B^3_{sys} 3

	Set 1	B^3_{sys}		
key	{a b c d e}	P	R	F
response$_1$	{a b c i j}	0.6	0.440	0.508
response$_2$	{a b c i j k}	0.5	0.440	0.468
response$_3$	{a b c i j k l}	0.429	0.440	0.434
response$_4$	{a b c i j k l m}	0.375	0.440	0.405

Table 6.7: Analysis of B^3_{sys} 4

6.1.3 *CEAF*

Luo (2005) criticizes the B^3 algorithm for using entities more than one time, because B^3 computes precision and recall of mentions by comparing entities containing that mention. Hence Luo proposes the *CEAF* algorithm which aligns entities in key and response. *CEAF* applies a similarity metric (which could be either based on mention or entity) for each pair of entities (i.e. a set of mentions) to measure the goodness of each possible alignment. The best mapping is used for calculating *CEAF* precision, recall and F-measure.

Consider the same example as cited for previous metrics,

Key : {m_1, m_2, m_3, m_4}

Response: {m_1, m_2} {m_3, m_4}

The best mapping of the key and response sets is illustrated in Figure 6.3. Since the response set R_1 is aligned with the key set K_1, R_2 is forced to align with an empty set.

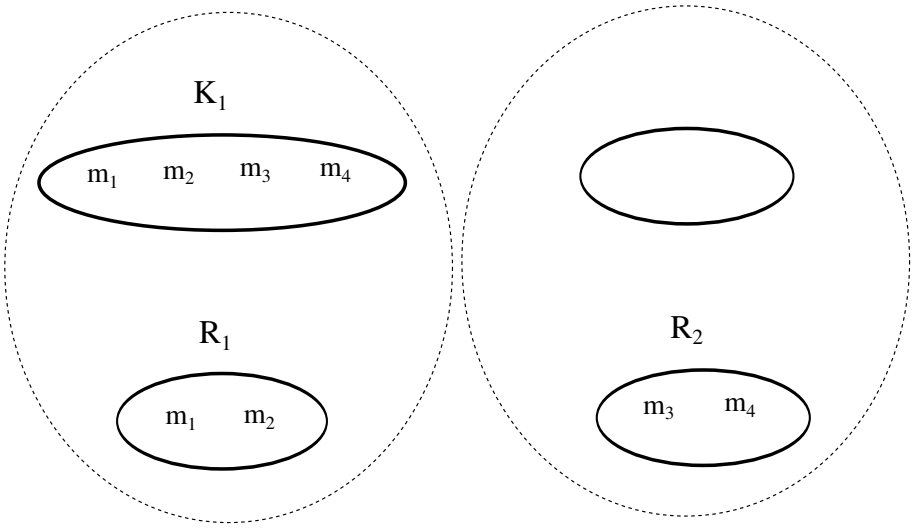

Figure 6.3: The *CEAF* Alignment Illustration

Luo proposes two entity-based similarity metrics (Equation 6.3 and 6.4) for an entity pair (K_i, R_j) originating from key, K_i, and response, R_j.

$$\phi_3(K_i, R_j) = |K_i \cap R_j| \tag{6.3}$$

$$\phi_4(K_i, R_j) = \frac{2|K_i \cap R_j|}{|K_i| + |R_j|} \tag{6.4}$$

The *CEAF* precision and recall are derived from the alignment which has the best total similarity (denoted as $\Phi(g^*)$), shown in Equations 6.5 and 6.6.

$$Precision = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \tag{6.5}$$

$$Recall = \frac{\Phi(g^*)}{\sum_i \phi(K_i, K_i)} \tag{6.6}$$

If not specified otherwise, we apply Luo's $\phi_3(\star, \star)$ in the example illustrations. We denote the original *CEAF* algorithm as *CEAF*$_{orig}$.

Detailed calculations are illustrated via a new example below:

Key : {a b c}

Response: {a b d}

The *CEAF*$_{orig}$ $\phi_3(\star, \star)$ are given by:

$\phi_3(K_1, R_1) = 2$ $(K_1 : \{abc\}; R_1 : \{abd\})$

$\phi_3(K_1, K_1) = 3$

$\phi_3(R_1, R_1) = 3$

So the *CEAF*$_{orig}$ evaluation numbers are:

$Pr_{CEAF_{orig}} = \frac{2}{3} = 0.667$

$Rec_{CEAF_{orig}} = \frac{2}{3} = 0.667$

$F_{CEAF_{orig}} = 2 \times \frac{0.667 \times 0.667}{0.667 + 0.667} = 0.667$

6.1.3.1 Problems of *CEAF*$_{orig}$

CEAF$_{orig}$ was intended to deal with key mentions. Its adaptation to system mentions has not been addressed explicitly. Although *CEAF*$_{orig}$ theoretically does not require the same number of mentions in key and response, it still cannot be directly applied to end-to-end systems, because the entity alignments are based on mention mappings.

As can be seen from Table 6.8, *CEAF*$_{orig}$ fails to produce intuitive results for system mentions. *System 2* outputs one more spurious entity (containing mention *i* and *j*) compared with *System 1*, however, achieves the same *CEAF*$_{orig}$ precision. Since twinless system mentions do

not have mappings in key, they contribute nothing to the mapping similarity. So, resolution mistakes for system mentions are not calculated, and moreover, the precision is easily skewed by the number of output entities. $CEAF_{orig}$ reports very low precision for system mentions (see also Stoyanov et al. (2009)).

		Set 1	Set 2	Singletons
System 1	key	{a b c}		
	response	{a b}		{c} {i} {j}
		P	R	F
$CEAF_{orig}$		0.4	0.667	0.500
B^3_{sys}		1.0	0.556	0.715
$CEAF_{sys}$		0.667	0.667	0.667
System 2	key	{a b c}		
	response	{a b}	{i j}	{c}
		P	R	F
$CEAF_{orig}$		0.4	0.667	0.500
B^3_{sys}		0.8	0.556	0.656
$CEAF_{sys}$		0.6	0.667	0.632

Table 6.8: Problems of $CEAF_{orig}$

		Set 1	Set 2	Set 3	Singletons
System 1	key	{a b c}			
	response	{a b}	{i j}	{k l}	{c}
		P	R	F	
$CEAF_{r\&n}$		0.286	0.667	0.400	
B^3_{sys}		0.714	0.556	0.625	
$CEAF_{sys}$		0.571	0.667	0.615	
System 2	key	{a b c}			
	response	{a b}	{i j k l}		{c}
		P	R	F	
$CEAF_{r\&n}$		0.286	0.667	0.400	
B^3_{sys}		0.571	0.556	0.563	
$CEAF_{sys}$		0.429	0.667	0.522	

Table 6.9: Problems of $\text{CEAF}_{r\&n}$

6.1.3.2 Existing *CEAF* variants

Rahman & Ng (2009) briefly introduce their *CEAF* variant, which is denoted as $CEAF_{r\&n}$ here. They use $\phi_3(\star,\star)$, which results in equal $CEAF_{r\&n}$ precision and recall figures when using true mentions. Since Rahman & Ng's experiments using system mentions produce unequal precision and recall figures, we assume that, after removing twinless singleton system mentions, they do not put any twinless mentions into the other set. In the example in Table 6.9, $CEAF_{r\&n}$ does not penalize adequately the incorrectly resolved entities consisting of twinless system mentions. So $CEAF_{r\&n}$ does not tell the difference between *System 1* and *System 2*. It can be concluded from the examples that the same number of mentions in key and response is needed for computing the *CEAF* score.

6.1.3.3 Our proposed variant — $CEAF_{sys}$

We propose to adjust *CEAF* in the same way as we did for B^3_{sys}, resulting in $CEAF_{sys}$. We put all twinless key mentions into the response as singletons. All singleton twinless system mentions are discarded. For calculating $CEAF_{sys}$ precision, all twinless system mentions which were mistakenly resolved are put into the key. For computing $CEAF_{sys}$ recall, only the

original key sets are considered. In this way *CEAF*$_{sys}$ deals adequately with system mentions (see Algorithm 5 for details).

Algorithm 5 $CEAF_{sys}$

Input: key sets key, response sets $response$

Output: precision P, recall R and F-score F

1: Discard all the singleton twinless system mentions in $response$;
2: Put all the twinless annotated mentions into $response$;
3: **if** calculating precision **then**
4: Merge all the remaining twinless system mentions with key to form key_p;
5: Use $response$ to form $response_p$
6: Form Map $g^\star$ between key_p and $response_p$
7: Calculate $CEAF$ precision P using $\phi_3(\star, \star)$
8: **end if**
9: **if** calculating recall **then**
10: Discard all the remaining twinless system mentions in $response$ to form $response_r$;
11: Use key to form key_r
12: Form Map $g^\star$ between key_r and $response_r$
13: Calculate $CEAF$ recall R using $\phi_3(\star, \star)$
14: **end if**
15: Calculate F-score F

Taking *System 2* in Table 6.8 as an example, key and response are altered for precision:

Key$_p$: {a b c} {i} {j}

Response$_p$: {a b d} {i j} {c}

So the $\phi_3(\star, \star)$ are as below, only listing the best mappings:

$\phi_3(K_1, R_1) = 2$ $(K_1 : \{abc\}; R_1 : \{abd\})$

$\phi_3(K_2, R_2) = 1$ $(K_2 : \{i\}; R_2 : \{ij\})$

$\phi_3(\emptyset, R_3) = 0$ $(R_3 : \{c\})$ $\phi_3(R_1, R_1) = 3$

$\phi_3(R_2, R_2) = 2$

$\phi_3(R_3, R_3) = 1$

The precision is thus given by:

$Pr_{CEAF_{sys}} = \frac{2+1+0}{3+2+1} = 0.6$

The key and response for recall are:

Key$_r$: {a b c}

Response$_r$: {a b} {c}

The resulting $\phi_3(\star, \star)$ are:

$\phi_3(K_1, R_1) = 2(K_1 : \{abc\}; R_1 : \{ab\})$

$\phi_3(\emptyset, R_2) = 0(R_2 : \{c\})$

$\phi_3(K_1, K_1) = 3$

$\phi_3(R_1, R_1) = 2$

$\phi_3(R_2, R_2) = 1$

The recall and F-score are thus calculated as:

$Rec_{CEAF_{sys}} = \frac{2}{3} = 0.667$

$F_{CEAF_{sys}} = 2 \times \frac{0.6 \times 0.667}{0.6+0.667} = 0.632$

However, one additional complication arises with regard to the similarity metrics used by *CEAF*. It turns out that only $\phi_3(\star, \star)$ is suitable for dealing with system mentions while $\phi_4(\star, \star)$ produces unintuitive results (see Table 6.10).

		Set 1	Singletons	
System 1	key	{a b c}		
	response	{a b}	{c} {i} {j}	
		P	R	F
$\phi_4(\star, \star)$		0.4	0.8	0.533
$\phi_3(\star, \star)$		0.667	0.667	0.667
System 2	key	{a b c}		
	response	{a b} {i j}	{c}	
		P	R	F
$\phi_4(\star, \star)$		0.489	0.8	0.607
$\phi_3(\star, \star)$		0.6	0.667	0.632

Table 6.10: Problems of $\phi_4(\star, \star)$

$\phi_4(\star,\star)$ computes a normalized similarity for each entity pair using the summed number of mentions in the key and the response. *CEAF* precision then distributes that similarity evenly over the response set. Spurious system entities, such as the one with mention *i* and *j* in Table 6.10, are not penalized. $\phi_3(\star,\star)$ calculates unnormalized similarities. It compares the two systems in Table 6.10 adequately. Hence we use only $\phi_3(\star,\star)$ in $CEAF_{sys}$.

When normalizing the similarities by the number of entities or mentions in the key (for recall) and the response (for precision), the *CEAF* algorithm considers all entities or mentions to be equally important. Hence *CEAF* tends to compute quite low precision for system mentions which does not represent the system performance adequately. Here, we do not address this issue.

6.1.4 *BLANC*

Recently, a new coreference resolution evaluation algorithm, *BLANC*, has been introduced (Recasens & Vila, 2010). This measure implements the *Rand index* (Rand, 1971) which has been originally developed to evaluate clustering methods. The *BLANC* algorithm deals correctly with singleton entities and rewards correct entities according to the number of mentions. However, a basic assumption behind *BLANC* is, that the sum of all coreferential and non-coreferential links is constant for a given set of mentions. This implies that *BLANC* assumes identical mentions in key and response. It is not clear how to adapt *BLANC* to system mentions. We do not address this issue here.

6.2 Experiments with the Proposed Evaluation Metrics

While Section 6.1 used toy examples to motivate our metrics B^3_{sys} and $CEAF_{sys}$, we here report results on two larger experiments using ACE2004 data.

6.2.1 Data and Mention Taggers

We use the ACE2004 (Mitchell et al., 2004) English training data which we split into three sets following Bengtson & Roth (2008): Train (268 docs), Dev (76), and Test (107). We use two in-house mention taggers. The first (*SM1*) implements a heuristic aiming at high recall. The second (*SM2*) uses the *J48* decision tree classifier (Witten & Frank, 2005). The number of detected mentions, head coverage, and accuracy on testing data are shown in Table 6.11.

		SM1	*SM2*
training	mentions	31,370	16,081
	twin mentions	13,072	14,179
development	mentions	8,045	–
	twin mentions	3,371	–
test	mentions	8,387	4,956
	twin mentions	4,242	4,212
	head coverage	79.3%	73.3%
	accuracy	57.3%	81.2%

Table 6.11: Mention Taggers on ACE2004 Data

6.2.2 The Artificial Setting

For the artificial setting we report results on the development data using the *SM1* tagger. To illustrate the stability of the evaluation metrics with respect to different mention taggers, we reduce the number of twinless system mentions in intervals of 10%, while correct (non-twinless) ones are kept untouched. The coreference resolution system used is the BART (Versley et al., 2008) reimplementation of Soon et al. (2001). The results are plotted in Figures 6.4 and 6.5.

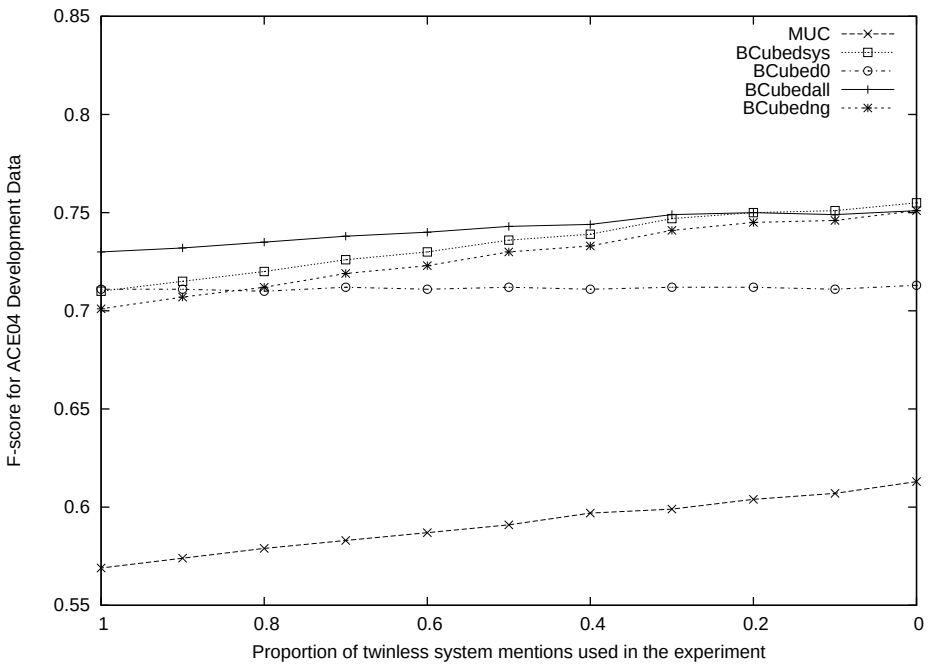

Figure 6.4: Artificial Setting B^3 Variants

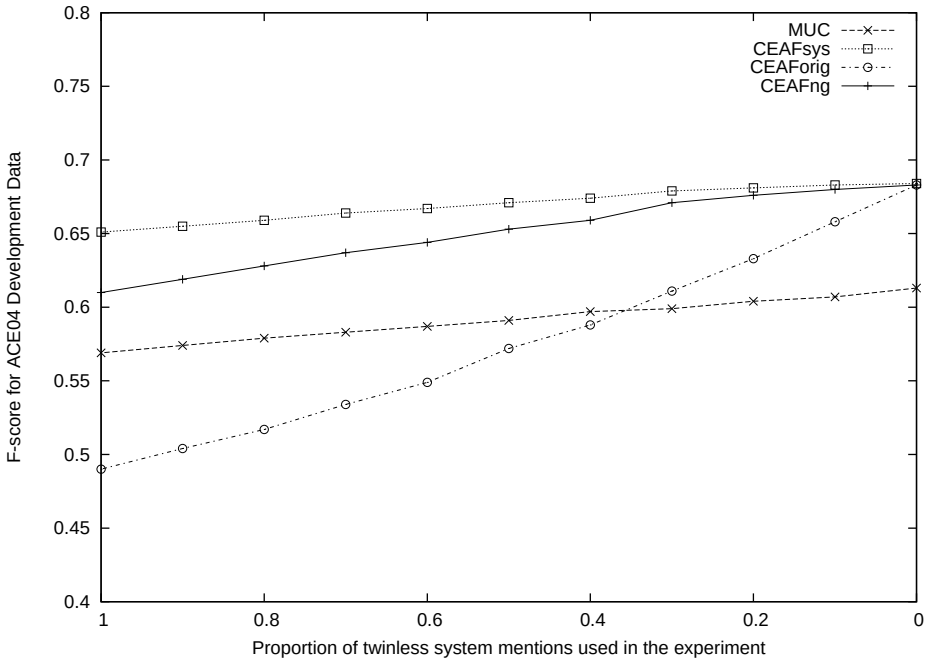

Figure 6.5: Artificial Setting *CEAF* Variants

Omitting twinless system mentions from the training data while keeping the number of correct mentions constant should improve the coreference resolution performance, because a more precise coreference resolution model is obtained. As can be seen from Figures 6.4 and 6.5, the *MUC*-score, B^3_{sys} and $CEAF_{sys}$ follow this intuition.

6.2.3 The Realistic Setting

Experiment 1 For the realistic setting we compare *SM1* and *SM2* as preprocessing components for the BART (Versley et al., 2008) reimplementation of Soon et al. (2001). The coreference resolution system with the *SM2* tagger performs better, because a better coreference model is achieved from system mentions with higher accuracy.

The *MUC*, B^3_{sys} and $CEAF_{sys}$ metrics have the same tendency when applied to systems with different mention taggers (Table 6.12, 6.13 and 6.14 and the bold numbers are higher with a p-value of 0.05, by a paired-t test). Since the *MUC* scorer does not evaluate singleton entities, it produces too low numbers which are not informative any more.

	MUC		
	R	Pr	F
Soon (SM1)	51.7	53.1	52.4
Soon (SM2)	49.1	69.9	**57.7**

Table 6.12: Realistic Setting *MUC*

	B^3_{sys}			B^3_0			B^3_{all}			$B^3_{r\&n}$		
	R	Pr	F	R	Pr	F	R	Pr	F	R	Pr	F
Soon (SM1)	65.7	76.8	70.8	57.0	91.1	**70.1**	65.1	85.8	74.0	65.1	78.7	71.2
Soon (SM2)	64.1	87.3	**73.9**	54.7	91.3	68.4	64.3	87.1	73.9	64.3	84.9	**73.2**

Table 6.13: Realistic Setting B^3 Variants

	CEAF$_{sys}$			*CEAF*$_{orig}$			*CEAF*$_{r\&n}$		
	R	Pr	F	R	Pr	F	R	Pr	F
Soon (SM1)	66.4	61.2	63.7	62.0	39.9	48.5	62.1	59.8	60.9
Soon (SM2)	67.4	65.2	**66.3**	60.0	56.6	**58.2**	60.0	66.2	62.9

Table 6.14: Realistic Setting *CEAF* Variants

As shown in Table 6.13, B^3_{all} reports counter-intuitive results when a system is fed with system mentions generated by different mention taggers. B^3_{all} cannot be used to evaluate two different end-to-end coreference resolution systems, because the mention tagger is likely to have bigger impact than the coreference resolution system. B^3_0 fails to generate the right comparison too, because it is too lenient by ignoring all twinless mentions.

The *CEAF*$_{orig}$ numbers in Table 6.14 illustrate the big influence the system mentions have on precision (e.g. the very low precision number for *Soon (SM1)*). The big improvement for *Soon (SM2)* is largely due to the system mentions it uses, rather than to different coreference models.

Both $B^3_{r\&n}$ and *CEAF*$_{r\&n}$ show no serious problems in the experimental results. However, as discussed before, they fail to penalize the spurious entities with twinless system mentions adequately.

	B^3_{sys}			B^3_0		
	R	Pr	F	R	Pr	F
Soon (SM2)	64.1	87.3	73.9	54.7	91.3	68.4
Bengtson	66.1	81.9	73.1	69.5	74.7	72.0

Table 6.15: Realistic Setting B^3_0 vs. B^3_{sys}

Experiment 2 We compare results of Bengtson & Roth's (2008) system with our *Soon (SM2)* system. Bengtson & Roth's embedded mention tagger aims at high precision, generating half of the mentions *SM1* generates (explicit statistics are not available to us).

Bengtson & Roth report a B^3 F-score for system mentions, which is very close to the one for true mentions. Their B^3-variant does not impute errors of twinless mentions and is assumed to be quite similar to the B^3_0 strategy.

We integrate both the B^3_0 and B^3_{sys} variants into their system and show results in Table 6.15 (we cannot report significance, because we do not have access to results for single documents in Bengtson & Roth's system). It can be seen that, when different variants of evaluation metrics are applied, the performance of the systems vary wildly.

6.3 Summary

In this chapter, we address problems of commonly used evaluation metrics for coreference resolution and suggest two variants for B^3 and *CEAF*, called B^3_{sys} and *CEAF*$_{sys}$. In contrast to the variants proposed by Stoyanov et al. (2009), B^3_{sys} and *CEAF*$_{sys}$ are able to deal with end-to-end systems which do not use any gold information. The numbers produced by B^3_{sys} and $CEAF_{sys}$ are able to indicate the resolution performance of a system more adequately, without being tricked easily by twisting preprocessing components. We believe that the explicit description of evaluation metrics, as given in this chapter, is a precondition for the reliable comparison of end-to-end coreference resolution systems.

Chapter 7

Evaluating *COPA*

In order to analyze the effectiveness of *COPA*, we present three groups of comparison experiments (1, 2, and 3) and two analytical ones (4 and 5) in this chapter.

1. Section 7.1 compares *COPA* against two baseline systems, both of which are pairwise models with strong features. The comparisons aim to convey the superiority of the global partitioning method proposed in *COPA* over local pairwise models, with all preprocessors (including the mention detector) being the same.

2. Section 7.2 shows the performance of *COPA* in the CoNLL 2011 shared task on coreference resolution, which is one of the most influential shared tasks in the field. Demonstrating *COPA*'s results in the task enables us to identify the competitiveness of our system, by comparing it with the most important state-of-the-art systems.

3. Section 7.3 tests *COPA* on medical data sets, to illustrate the robustness of *COPA* when adapted to new domains.

4. Experiments on the weakly supervised property of *COPA* are shown in Section 7.5.

5. Experiments on analyzing our proposed k *model* are in Section 7.6.

Since the experimental settings differ between sections, discussions are provided separately in each section, making them self-contained. Features mentioned in this chapter are described in Chapter 5 in more details, and the data sets are introduced in Chapter 3.

7.1 *COPA* vs. Baselines

We compare *COPA* with two implementations of pairwise models. The first baseline is *SOON* – the BART (Versley et al., 2008) reimplementation of Soon et al. (2001), with few (i.e. 12)

but strong features. Our second baseline is *B&R* – Bengtson & Roth (2008) [1], which exploits a much larger feature set while keeping the machine learning approach simple. Bengtson & Roth (2008) show that their system outperforms much more sophisticated machine learning approaches such as Culotta et al. (2007), who reported the best results on true mentions before Bengtson & Roth (2008). Bengtson & Roth (2008)'s is the strongest pairwise model on the ACE data sets before the CoNLL 2011 shared task (which is discussed in Section 7.2), and its source code is accessible for modifications so that strict fair comparisons can be conducted. Therefore Bengtson & Roth (2008)'s system is the second reasonable competitor for evaluating *COPA* in this Section.

Both of the baseline systems are chosen because they are the strongest pairwise models to compare with to illustrate the effectiveness of our proposed global method. We use **the same pre-processors (including the mention detection)** for all systems to exclude the possible influences from them. Differences in outputs mainly indicates the differences in the inference algorithms.

7.1.1 Data

We use the MUC6 data (Chinchor & Sundheim, 2003) with the standard training/testing divisions (30/30) and the MUC7 data (Chinchor, 2001) (30/20). Since we do not have access to the official ACE testing data (only available to ACE participants), we follow Bengtson & Roth (2008) for dividing the ACE 2004 English training set (Mitchell et al., 2004) into training, development and testing partitions (268/76/107). We randomly split the 252 ACE 2003 training documents (Mitchell et al., 2003) using the same proportions into training, development and testing (151/38/63). The systems were tuned on development data and run only once on testing data.

7.1.2 The Mention Tagger

We implement a classification-based mention tagger, which tags each NP chunk (e.g. the output of the Yamcha Chunker) as being an ACE mention or not, with the necessary post-processing for embedded mentions. For the ACE 2004 testing data, we cover 75.8% of the syntactic heads of mentions with a 73.5% accuracy.

Since the MUC data sets do not limit the mentions to any specific semantic classes as the ACE sets do, our mention tagger directly outputs all the embedded noun phrases.

[1] `http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=FLBJCOREF`

7.1.3 Evaluation Metrics

In order to report realistic results, we neither assume true mentions as input nor do we evaluate only on true mentions. Instead, we use an in-house mention tagger for automatically extracting mentions, and evaluate using variants of the evaluation metrics B^3 (Bagga & Baldwin, 1998) and *CEAF* (Luo, 2005), named B^3_{sys} and *CEAF*$_{sys}$ respectively, which are adapted to the evaluation of end-to-end coreference resolution systems (see Chapter 6). For the sake of completeness we also report the *MUC* score.

7.1.4 Results

7.1.4.1 *COPA* vs. *SOON*

In this section, we compare the *SOON*-baseline with *COPA* using the *R2 partitioner* (parameters $\alpha^\star$ and β optimized on development data). *COPA* uses the same features as adopted by *SOON*, which are shown in Table 7.1. Moreover, the two systems use the same set of system mentions too.

Negative	(1) N_Gender, (2) N_Number, (3) N_SemanticClass
Positive	(10) StrMatch_Npron, (11) StrMatch_Pron, (12) Alias, (14) Nprn_Prn, (21) Appositive, (31)sentence distance

Table 7.1: *COPA* Features for Comparing with *SOON* (details in Chapter 5)

Table 7.2 gives the comparison results, it can be seen that even with the same features, *COPA* consistently outperforms *SOON* on all data sets using all evaluation metrics. With the exception of MUC7, ACE 2003 and ACE 2004 data evaluated with *CEAF*$_{sys}$, all of *COPA*'s improvements are statistically significant. When evaluated using *MUC* and B^3_{sys}, *COPA* with the *R2 partitioner* boosts recall in all data sets while losing in precision. This led us to believe that incorporating more features would increase precision without losing too much recall. Hence we integrated features from Bengtson & Roth (2008)'s system to conduct the second comparison in Section 7.1.4.2.

		SOON			COPA with the *R2 partitioner*				
		R	P	F	R	P	F	$\alpha^\star$	β
MUC	MUC6	59.4	67.9	63.4	62.8	66.4	**64.5**	0.08	0.03
	MUC7	52.3	67.1	58.8	55.2	66.1	**60.1**	0.05	0.01
	ACE 2003	56.7	75.8	64.9	60.8	75.1	**67.2**	0.07	0.03
	ACE 2004	50.4	67.4	57.7	54.1	67.3	**60.0**	0.05	0.04
B^3_{sys}	MUC6	53.1	78.9	63.5	56.4	76.3	**64.1**	0.08	0.03
	MUC7	49.8	80.0	61.4	53.3	76.1	**62.7**	0.05	0.01
	ACE 2003	66.9	87.7	75.9	71.5	83.3	**77.0**	0.07	0.03
	ACE 2004	64.7	85.7	73.8	67.3	83.4	**74.5**	0.07	0.03
$CEAF_{sys}$	MUC6	56.9	53.0	54.9	62.2	57.5	**59.8**	0.08	0.03
	MUC7	57.3	54.3	55.7	58.3	54.2	56.2	0.06	0.01
	ACE 2003	71.0	68.7	69.8	71.1	68.3	69.7	0.07	0.03
	ACE 2004	67.9	65.2	66.5	68.5	65.5	67.0	0.07	0.03

Table 7.2: *SOON* vs. *COPA* R2 (*SOON* features, system mentions, bold indicates significant improvement in F-score over *SOON* according to a paired-t test with $p < 0.05$)

In brief, Table 7.2 conveys that the global hypergraph partitioning method of *COPA* models the coreference resolution task more adequately than Soon et al. (2001)'s local model – even when using the very same features and the same mentions.

7.1.4.2 *COPA* vs. *B&R*

Table 7.3 gives our re-produced *B&R* numbers on the ACE 2004 testing data using the true (and system) mention settings, in comparison to the numbers they reported in the paper. Their lenient variant of B^3 (Stoyanov et al., 2009) is used, which discards all twinless mentions[2]. Table 7.3 is to show that we make sure that their reported numbers are successfully regenerated. Replacing their preprocessing components with ours generates 74.8 F-score of B^3_{sys}, which is comparable to the 74.0 using their own's.

[2]The mentions which are not aligned with true mentions are called twinless (Stoyanov et al., 2009)

	Reported			**Reproduced**		
	R	P	F	R	P	F
true mention (lenient B^3)	74.5	88.3	80.8	73.0	89.6	80.4
B&R's system mention (lenient B^3)	72.5	84.9	78.24	72.1	83.2	77.3
B&R's system mention (B^3_{sys})	-	-	-	68.3	80.8	74.0
COPA's system mention (B^3_{sys})	-	-	73.8	66.3	85.8	74.8

Table 7.3: Reproduced Numbers of *B&R*

In Table 7.4 we report the B^3_{sys} performance of *SOON* and *B&R* on the ACE 2004 testing data (which was the data set *B&R*'s original results reported on) using true mentions and using *COPA*'s automatically identified system mentions. For evaluation we use B^3_{sys} only, because (Bengtson & Roth, 2008)'s system does not allow one to easily integrate *CEAF*. *B&R* considerably outperforms *SOON* (we cannot compute statistical significance, because *B&R* does not provide single document performance). The difference using system mentions, however, is not as big as we expected. Bengtson & Roth (2008) reported very good results when using true mentions. For evaluating on system mentions, however, they were using the lenient B^3. When replacing this with B^3_{sys} the difference between *SOON* and *B&R* shrinks.

	SOON			*B&R* (Reproduced)		
	R	P	F	R	P	F
true mention (B^3_{sys})	67.4	90.3	77.2	73.0	89.6	80.4
COPA's system mention (B^3_{sys})	64.7	85.7	73.8	66.3	85.8	74.8

Table 7.4: Baselines on the ACE 2004 Testing Data

In this section, we compare the *B&R* system (using our preprocessing components and mention tagger), and *COPA* with the *R2 partitioner* using *B&R* features. The features are given in Table 7.5. *COPA* does not use the learned features from *B&R*, as this would have implied to embed a pairwise coreference resolution system in *COPA*.

Negative	(1) N_Gender, (2) N_Number, (3) N_SemanticClass (4) N_Mod,
Positive	(10) StrMatch_Npron, (11) StrMatch_Pron, (12) Alias, (13) HeadMatch,(14) Nprn_Prn, (21) Appositive, (31)sentence distance, (32) compatible mention distance
Weak	(27) W_VerbAgree, (29) W_Synonym

Table 7.5: *COPA* Features for Comparing with *B&R* (details in Chapter 5)

The comparison results are provided in Table 7.6. We report results for ACE 2003 and ACE 2004. The parameters are optimized on the ACE 2004 data. *COPA* with the *R2 partitioner* outperforms *B&R* on both data sets. Bengtson & Roth (2008) developed their system on ACE 2004 data and never exposed it to ACE 2003 data. We suspect that the relatively poor result of *B&R* on ACE 2003 data is caused by its over-fitting to ACE 2004. This shows that *COPA* is a highly competitive system as it outperforms Bengtson & Roth (2008)'s system which claims to have the best performance on the ACE 2004 data.

		B&R			*COPA* with the*R2 partitioner*		
		R	P	F	R	P	F
B^3_{sys}	ACE 2003	56.4	97.3	71.4	70.3	86.5	77.5
	ACE 2004	66.3	85.8	74.8	68.4	84.4	75.6

Table 7.6: *B&R* vs. *COPA* R2 (*B&R* features, *COPA*'s system mentions)

7.1.4.3 Running Time

On a machine with 2 AMD Opteron CPUs and 8 GB RAM, *COPA* finishes preprocessing, training and partitioning the ACE 2004 data set in 15 minutes, which is slightly faster than our duplicated *SOON* baseline and is much faster than the original *B&R* system.

7.1.5 Discussion

Most previous attempts to solve the coreference resolution task globally have been hampered by employing a local pairwise model in the classification step (i.e. step 1 mentioned in Chapter

2) while only the clustering step realizes a global approach (E.g. Luo et al. (2004), Nicolae & Nicolae (2006), Klenner (2007), Denis & Baldridge (2009), lesser so Culotta et al. (2007)). In this section, we conduct experiments comparing our coreference resolution system, *COPA*, against two strong baselines (Soon et al., 2001; Bengtson & Roth, 2008). Soon et al. (2001) is the first two-step model with 12 very strong features. Bengtson & Roth (2008)'s system has been claimed to achieve the best performance on the ACE 2004 data (using true mentions, Bengtson & Roth (2008) did not report any comparison with other systems using system mentions). *COPA* implements a global decision in one step via hypergraph partitioning and considers all the relations in a graph, which enables it to **outperform the two strong pairwise models**.

It has been observed that the improved performance with true mentions do not necessarily translate to an improved performance when system mentions are used (Ng, 2008). We follow Stoyanov et al. (2009) and argue that evaluating the performance of coreference resolution systems on true mentions is unrealistic. Hence we integrate an ACE mention tagger into our system, tune the system towards the real task, and evaluate only using system mentions. While Ng (2008) could not show that superior models achieved superior results on system mentions, *COPA* is able to outperform both baseline systems **in strict comparisons and in an end-to-end setup**.

7.2 *COPA* vs. State-of-the-art Systems

COPA has participated in the CoNLL shared task on modeling unrestricted coreference (Pradhan et al., 2011), and we submitted *COPA*'s results to the *open* setting of the task. We used only 30% of the training data (randomly selected) and 20 features (see Table 7.7).

Negative	(1) N_Gender, (2) N_Number, (3) N_SemanticClass, (4) N_Mod, (5) N_DSPrn, (6) N_ContraSubjObj
Positive	(10) StrMatch_Npron, (11) StrMatch_Pron, (12) Alias, (13) HeadMatch, (14) Nprn_Prn, (15) Speaker12Prn, (16) DSPrn, (17) ReflexivePrn, (18) PossPrn, (19) GPEIsA, (20) OrgIsA, (31) sentence distance (32) compatible mention distance
Weak	(27) W_VerbAgree, (28) W_Subject, (29) W_Synonym

Table 7.7: *COPA* Features for the CoNLL 2011 Shared Task (details in Chapter 5)

7.2.1 Data

The CoNLL shared task aims to predict coreference on the OntoNotes data. There are 1,674 training documents, 202 development documents and 207 testing documents. As is customary for CoNLL tasks, two tracks are provided, i.e. closed and open. For the closed track, participating systems are restricted to using the distributed resources (with the predicted layers of information provided by the task), in order to allow fair algorithmic comparisons. The open track allows for unrestricted usage of additional external resources. Since several off-the-shelf pre-processing components are used, *COPA* participates in the open setting track (without actually using additional resources such as Wikipedia).

7.2.2 The Mention Tagger

For the CoNLL shared task, we incorporate information from syntactic parse trees into our mention tagger. Both the semantic classes and the syntactic heads are generated along with the system mentions. The official evaluation on the mention taggers shows that the performance of our mention tagger falls into the average-performance group (see Table 7.8).

	R	P	F1
COPA	67.15	67.64	67.40
max open	74.31	67.87	70.94

Table 7.8: *COPA*'s Mention Tagger Performance on the CoNLL testing set

7.2.3 Evaluation Metrics

The unweighted average of *MUC*, *BCUBED* and *CEAF(E)* is used as the final score in CoNLL shared task. *CEAF(E)* is using the entity based similarity metric (see Chapter 6). It is considered that each of the three metrics represents a different important dimension (Denis & Baldridge, 2009), the *MUC* being based on links, *BCUBED* based on mentions and *CEAF* on entities. The combination of them should be adequate for evaluating the performances of a coreference resolution system.

7.2.4 Results

The stopping criterion $\alpha*$ (see Section 4.2.2.2) is tuned on development data to optimize the final coreference scores. A value of 0.06 is chosen for the CoNLL testing set.

COPA's results on the development set and the testing set are displayed in Table 7.9 and Table 7.10 respectively. The *Overall* numbers in both tables are the average scores of *MUC*, *BCUBED* and *CEAF(E)*. In Table 7.11, the best performances in both open and closed are given, along with the median numbers. Since *COPA* is not using additional resources anyway, the closed numbers can still be roughly compared with. This is mentioned in the overview paper of the task too (see the second paragraph in page 18 of (Pradhan et al., 2011)).

Metric	R	P	F1
MUC	52.69	57.94	55.19
BCUBED	64.26	73.39	68.52
CEAF(M)	54.44	54.44	54.44
CEAF(E)	45.73	40.92	43.19
BLANC	69.78	75.26	72.13
Overall			55.63

Table 7.9: *COPA*'s results on the CoNLL development set

Metric	R	P	F1
MUC	56.73	58.90	57.80
BCUBED	64.60	71.03	67.66
CEAF(M)	53.37	53.37	53.37
CEAF(E)	42.71	40.68	41.67
BLANC	69.77	73.96	71.62
Overall			55.71

Table 7.10: *COPA*'s results on the CoNLL testing set

	F1
COPA	55.71
max open	58.31
med open	54.32
max closed	57.79
med closed	50.98

Table 7.11: Overall Results on the CoNLL testing set

The best system of CoNLL 2011 shared task is Stanford's Multi-Pass Sieve system (Lee et al., 2011), which is based on heuristic rules. The second ranking systems are not significantly different from ours, for instance Sapena's system, which uses an iterative probabilistic model with the constraints between mentions learned from a decision tree. Both of the systems are described in Chapter 2. Overall, *COPA* performs competitively when compared with the state-of-the-art systems in the field, while using a relatively small set of features and a small amount of training data.

7.2.5 Discussions

The CoNLL 2011 shared task enables us to compare our coreference model *COPA* with the state-of-the-art systems on a much bigger data set, the OntoNotes data. We only apply **30% of the training documents** to learn the hyperedge weights, and the learned *COPA* model comes in as **the second team** in the open track in which five teams participated. Since *COPA* does not use additional resources, it is considered to belong to the second small ball park in the closed track too (Pradhan et al., 2011) where there are 18 teams participating.

Pradhan et al. (2011) concludes that most of the participating systems are still two-step models, fully trained upon the training set using the approach as described in (Soon et al., 2001). It is suggesting again that *COPA*'s global partitioning algorithm **outperforms the pairwise models under the CoNLL setup**, even with a small set of features (i.e. 22).

7.3 *COPA* in the Medical Domain

We participated in all three tasks of the 2011 i2b2/VA Track on Challenges in Natural Language Processing for Clinical Data (descriptions can be found in Chapter 3). The features used to report the results are given in Table 7.12.

Negative	(1) N_Gender, (2) N_Number, (3) N_SemanticClass, (4) N_Mod, (6) N_ContraSubjObj, (7) N_i2b2Type, (8) N_i2b2Quant, (9) N_i2b2ConName
Positive	(10) StrMatch_Npron, (11) StrMatch_Pron, (12) Alias, (13) HeadMatch, (14) Nprn_Prn, (17) ReflexivePrn, (21) Appositive, (23) i2b2PisA, (24) i2b2Abbr, (25) i2b2CatMatch, (26) i2b2PronPreference, (31)sentence distance, (32) compatible mention distance
Weak	(28) W_Subject,(29) W_Synonym, (30) W_i2b2SubStr

Table 7.12: *COPA* Features for the 2011 i2b2/VA Shared Task (details in Chapter 5)

7.3.1 Data

For task 1A and task 1B – ODIE corpus without and with concepts[3], a training set of 97 documents is released (including the Mayo and Pittsburgh data sets). A total number of 492 documents (including the Partner, Beth and Pittsburgh data sets) are used as training data for task 1C – i2b2/VA corpus with concepts. In task 1A, our in-house mention tagger is integrated into the preprocessing components.

For development purposes, we randomly split the training data into two parts with the ratio of 4 to 1. From the ODIE corpus, 78 documents are kept for training, and 19 are used as development set. A split of 394/98 is used for the i2b2/VA corpus.

7.3.2 The Mention Tagger

For the I2B2 shared task, the semantic classes of mentions (e.g. persons and treatments) are evaluated together with the output coreference sets in task 1A. Our mention tagger makes use of the entity definitions extracted from the Unified Medical Language System (UMLS) [4] for the semantic class identification. Our mention tagger covers 84.9% of the syntactic heads of mentions with an accuracy of 62.2% on the ODIE corpus.

[3]Concepts in the shared task refer to the given true mentions.

[4]`http://www.nlm.nih.gov/research/umls/`

7.3.3 Evaluation Metrics

For coreference resolution there exists no evaluation metric that has been approved unanimously. Hence the i2b2/VA/Cincinnati shared task adopts the approach taken by the CoNLL 2011 shared task to measure the final coreference performance, the unweighted average of the *MUC*, *BCUBED* and *CEAF(E)* evaluation metrics, here being denoted as *Overall*. However, in contrast to the *CoNLL* evaluation, the i2b2/VA/Cincinnati shared task evaluates additional mentions that do not participate in any coreference set, so that it results in too high performance numbers (see *BCUBED* numbers in Table 7.15 for an example). In addition, i2b2/VA/Cincinnati adopts the *BLANC* evaluation metric but does not include it in *Overall*. We report numbers according to the i2b2/VA/Cincinnati evaluation scripts for Task 1B and Task 1C (denoted as ***I2B2***). For task 1A (with automatically detected mentions) we compute the evaluation metrics according to our own variants of *BCUBED* and *CEAF* (denoted as ***SYS***), and CoNLLs variants of *BCUBED* and *CEAF* (denoted as ***CoNLL***). Reporting our results for task 1A using the *I2B2* metrics is meaningless because the final i2b2/VA/Cincinnati evaluation script also evaluates the semantic classes of mentions which we do not include into our output files. The final i2b2/VA/Cincinnati evaluation script changed during the final evaluation phase. The released script during the development phase actually does not evaluate the semantic classes. All evaluations in this section are conducted across semantic classes.

7.3.4 Results

***COPA* on the Development Data.** *COPA*'s results on the development sets for all three tasks are displayed in Table 7.13, Table 7.14, Table 7.15 and Table 7.16. The evaluation metrics (i.e. *MUC*, *BCUBED*, *CEAF(E)*, *overall* as the unweighted average of the three, and additionally *BLANC*) are calculated with the scripts provided by the shared task.

task 1A (*SYS*)	**R**	**P**	**F1**
MUC	88.9	61.8	72.9
BCUBED	83	90	86.4
CEAF	78.5	63.6	70.2
Overall			**76.5**

Table 7.13: *COPA*'s Results on the ODIE Development Set w/o Concepts (Task 1A) Using *SYS* Evaluation Metrics

task 1A (*CoNLL*)	**R**	**P**	**F1**
MUC	88.9	61.8	72.9
BCUBED	82.5	94.4	88
CEAF	78.5	48.2	59.7
Overall			**73.6**

Table 7.14: *COPA*'s Results on the ODIE Development Set w/o Concepts (Task 1A) Using *CoNLL* Evaluation Metrics

task 1B (*I2B2*)	**R**	**P**	**F1**
MUC	88.6	79.1	82.7
BCUBED	88.5	93	90.7
CEAF	71.5	62.2	66.5
(*BLANC*	80.5	95.8	86.6)
Overall			**80.0**

Table 7.15: *COPA*'s Results on the ODIE Development Set with Concepts (Task 1B) Using *I2B2* Evaluation Metrics

task 1C (*I2B2*)	**R**	**P**	**F1**
MUC	80.8	84.9	82.8
BCUBED	95.6	96.1	95.8
CEAF	88.8	86.3	87.6
(*BLANC*	93.3	97.2	95.2)
Overall			**88.7**

Table 7.16: *COPA*'s Results on the i2b2/VA Development Set with Concepts (Task 1C) Using *I2B2* Evaluation Metrics

***COPA* on the Testing Data.** Our final performances on the testing data for Task 1B (i.e. overall F1 measure of 0.806) and Task 1C (i.e. overall F1 measure of 0.888) are similar to our results on the development set (see Table 7.15 and Table 7.16).

Our testing results are slightly worse than the results of the top performing system for Task 1C, and are not significantly different from the top results for Task 1B (Uzuner et al., 2012). It is indicating that our system is competitive in the medical domain. However, our results on the testing data of Task 1A are much worse than on the development data, because the final evaluation script (*I2B2*) also evaluates the semantic classes of mentions too, which we did not include into our output files. It can be seen from Table 7.17 that, *SYS* metrics give similar numbers on the Task 1A testing data as on the Task 1A development data, which are the best *SYS* performances in the shared task.

task 1A (*SYS*)	**R**	**P**	**F1**	**F1 max**	**F1 med**
Exact and Partial	.760	.648	**.696**	.696	.690
Exact	.783	.707	**.730**	.730	.703

Table 7.17: *COPA*'s Results (in bold) on the ODIE Testing Set w/o Concepts (Task 1A) Using *SYS* Evaluation Metrics

task 1A (*I2B2*)	**R**	**P**	**F1**	**F1 max**	**F1 med**
Exact and Partial	.617	.423	**.417**	.657	.624
Exact	.765	.568	**.630**	.675	.634

Table 7.18: *COPA*'s Results (in bold) on the ODIE Testing Set w/o Concepts (Task 1A) Using *I2B2* Evaluation Metrics

task 1B (*I2B2*)	**R**	**P**	**F1**	**F1 max**	**F1 med**
Overall	.850	.773	**.806**	.827	.800

Table 7.19: *COPA*'s Results (in bold) on the ODIE Testing Set with Concepts (Task 1B) Using *I2B2* Evaluation Metrics

task 1C (*I2B2*)	**R**	**P**	**F1**	**F1 max**	**F1 med**
Overall	.894	.882	**.888**	.915	.859

Table 7.20: *COPA*'s Results (in bold) on the i2b2/VA Testing Set with Concepts (Task 1C) Using *I2B2* Evaluation Metrics

Medical Domain Knowledge. As mentioned in Chapter 5, the UMLS thesaurus and the MetaMap API are used to equip *COPA* with medical domain knowledge. Features **(7) N_i2b2Type**, **(9) N_i2b2ConName**, **(23) i2b2PisA** and **(24) i2b2Abbr** are left out in Table 7.21 to illustrate the influence of domain knowledge.

	w/o KnowledgeFeats			**w KnowledgeFeats**		
task 1C(*I2B2*)	**R**	**P**	**F1**	**R**	**P**	**F1**
MUC	.807	.821	.814	.808	.849	**.828**
BCUBED	.959	.953	.956	.956	.961	.958
CEAF	.859	.867	.863	.888	.863	**.876**
Overall			.878			**.887**

Table 7.21: *COPA*'s Results on the i2b2/VA Development Set with Concepts (Task 1C), with and without Knowledge Features, Using *I2B2* Evaluation Metrics. (bold indicates significant improvement in F1 measure over the column w/o KnowledgeFeats, according to a paired-t test with $p < 0.005$)

By accessing domain knowledge, *COPA* manages to capture the coreference relation which pure linguistic features cannot capture. For example, the mention {*neurolysis*} is correctly resolved to {*the procedure*}*R* due to the contribution of the *IsA* relation. Because the version of the evaluation metrics used by the shared task is overwhelmed by unresolved singletons (in particular *BCUBED*), the contribution of the knowledge features appears smaller than it actually is. The same comparison is conducted with *SYS* metrics in Table 7.22, which shows a bigger improvement by using knowledge features.

	w/o KnowledgeFeats			w KnowledgeFeats		
task 1C (*SYS*)	**R**	**P**	**F1**	**R**	**P**	**F1**
MUC	.807	.821	.814	.808	.849	**.828**
BCUBED	.750	.849	.797	.752	.883	**.813**
CEAF	.786	.731	.757	.792	.750	**.770**
Overall			.787			**.804**

Table 7.22: *COPA*'s Results on the i2b2/VA Development Set with Concepts (Task 1C), with and without Knowledge Features, Using *SYS* Evaluation Metrics. (bold indicates significant improvement in F1 measure over the column w/o KnowledgeFeats, according to a paired-t test with $p < 0.005$)

7.3.5 Discussions

By participating in the I2B2 shared task, we are able to convey the **domain adaptation** ability of the *COPA* model. With the system mention setting and the *SYS* metrics (see Table 7.17), *COPA* generates the **best performance**. In terms of the true mention setting, *COPA* is ranked into the **second group** (Uzuner et al., 2012).

From the experiences in the I2B2 shared task, we confirm that it is easy to adapt the *COPA* model to new domains. The **feature engineering is easy** due to the overlapping hyperedges and the **learning phase can be cheaply done** with a small portion of the training documents.

7.4 Error Analysis

7.4.1 *COPA* Errors for News Articles

Mention Detection Errors. As described in Section 4.3.1, our mention detection is based on automatically extracted information, such as syntactic parsing trees and basic NP chunks. Since no *minimum span* information is provided in the OntoNotes data (in contrast to the previous standard corpus, ACE), exact mention-boundary detection is required. A lot of the spurious mentions in our system are generated due to the mismatches of the ending or starting punctuations, and the OntoNotes annotation is also not consistent in this regard. The mention detection F-score of *COPA* is 67.40, whereas the best system in the CoNLL shared task has the F-score of 70.94.

Our current mention detector does not extract verb phrases. Therefore it misses all the *Event* mentions in the OntoNotes corpus.Besides the fact that the current *COPA* is not resolving any *event coreferences*, our mention detector performs weakly in extracting *date* mentions too. As a result, the system outputs several spurious coreference sets, for instance a set containing the *September* from the mention *15th September*. Moreover, an idiomatic expression identification needs to be included too, which should help to avoid detecting some spurious mentions, such as {*God*} in the phrase {*for God's sake*}.

Resolution Errors. A big portion of the recall loss in our system is due to the lack of world knowledge. For example, *COPA* does not resolve the mention {*the Europe station*} correctly into the entity RADIO FREE EUROPE, because the system does not know that the entity is a station.

Some more difficult coreference cases in the *OntoNotes* data might require a reasoning mechanism. To be able to connect the mention {*the victim*} with the mention {*the groom's brother*}, the event that the brother is killed needs to be interpreted by the system.

We also observed from the experiments that the resolution of the {*it*} mentions are quite inaccurate. Although our mention detector discards the pleonastic pronouns, there are still a lot of them left that introduce wrong coreference sets. Since the {*it*} mentions do not contain enough information by themselves, more features exploring their local syntax are necessary.

7.4.2 *COPA* Errors for Clinical Reports

The data sets adopted in the i2b2/VA shared task contain semi-structured reports describing clinical relevant information of patients. Therefore some data-specific coreference chains can be easily derived, such as in the case of "{*Patient*} name: {*XXX*}" where the patient name is explicitly given. Pronouns in these data sets are not as ambiguous as they are in news articles. The patient is quite centered in the context of each report, who occupies most of the third person pronouns. Most singular first person pronouns refer to the doctors who write the reports.

Definite noun phrases are not used frequently in the i2b2/VA data sets. Instead, variations of medical terms and expanded descriptions of entities frequently appear, which are difficult to detect without domain-dependent knowledge resources.

Mention Detection Errors. The mention detection in task 1A has been a challenge for us, as the annotated mentions are not always the largest noun phrase spans (which is usually the case in coreference annotations). Annotated is rather a meaningful medical usage. For instance, phrase {*appendix 8.0 x 0.5 cm*} is a mention while {*135 pulse rate*} is not.

Resolution Errors. *COPA* has difficulties deciding whether the difference between the modifiers of the mention {*chest pain*} and the mention {*back pain*} is essential enough to separate them from each other. It requires knowledge that {*back*} and {*chest*} are both part of the body while being different ones. We attempt to handle this problem by including the medical concept names the mentions refer to (see feature (9)). However, including even deeper knowledge would be beneficial.

7.5 Experiments on the Training Data Size

We conducted a series of runs with different amounts of the training data, shown in Figure 7.1. The curve derived from the i2b2/VA/Cincinnati corpus using the *I2B2* metrics is tagged with "i2b2_trsize", while the curve using our *SYS* metrics is tagged with "i2b2_trsize,sys". Because of the skewed evaluation metrics adopted in the i2b2/VA/Cincinnati (see Section 7.3.3), the curve "i2b2_trsize" shows only a small drop in performance (i.e. four percent F-measure) when only two training documents are used. When we apply our own version of the evaluation metrics which is not as influenced by singletons (see Chapter 6), the drop on the curve "i2b2_trsize,sys" is more pronounced. However, even with this evaluation measure we can see that only little training data is sufficient for our system to reach its top performance.

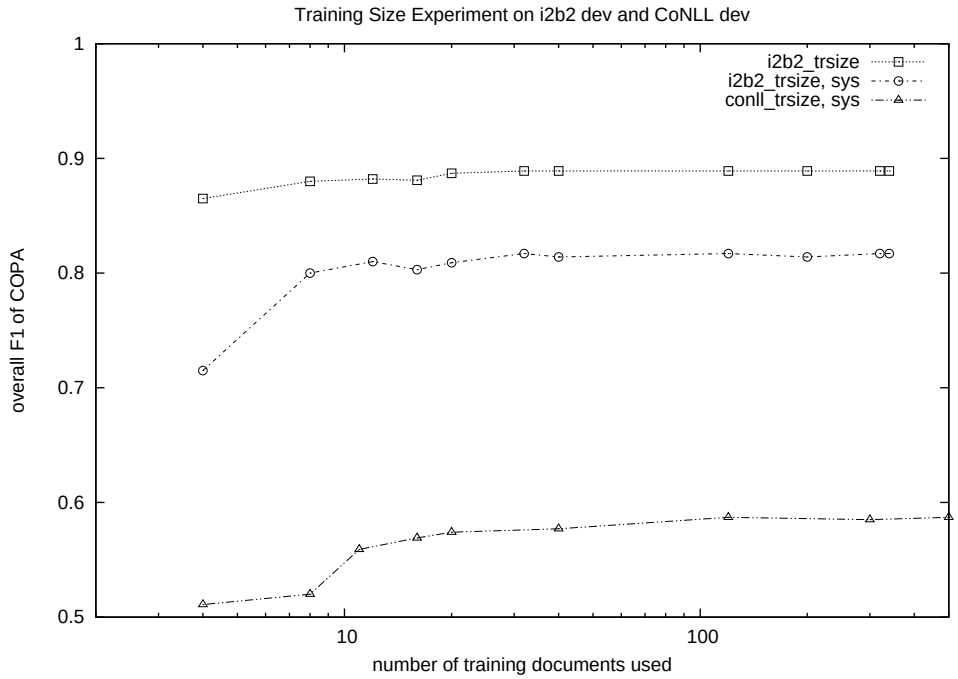

Figure 7.1: *COPA*'s Results with Different Sizes of the Training Data

In order to check whether the task of coreference resolution is easier in the clinical domain than in the news domain, we perform the same experiment using the CoNLL-shared task development data using our own evaluation metrics ("sys"), the curve of which is tagged as "conll_trsize,sys". Here we see a slight increase when using more than 20 training documents, though even here we reach top performance with only about 100 training documents (out of more than 1,800 original ones). The overall lower numbers can be partially explained by using automatically tagged mentions and partially by the difficulty of the news domain (due to the more occurrences of pronouns and diverse entity types). However, in both domains our system needs only very little training data to achieve competitive performance.

7.6 Experiments on the k Model

We proposed two partitioning algorithms in this thesis, the *R2 partitioner* which partitions the hypergraphs in an iterative manner and the *flatK partitioner* which attempts to conquer the hierarchical limitation of the *R2 partitioner* by deriving the clusters at one step. The *flatK partitioner* assumes the number of clusters to be known beforehand, and our proposed k *model* in Chapter 4 addresses this issue via preference modeling.

The effect of singleton entities. It is no trivial matter to predict the number of entities (i.e. clusters) during the end-to-end coreference processing, when noise is involved in the graphs to be partitioned. System mentions which do not participate in any coreference set present as singleton entities in the graphs, which dramatically change the distributions of the number of entities.

Figure 7.2 compares the distributions of the number of entities per 100 mentions with and without singleton entities involved. The figures on the left side plot the frequencies of different k's without singleton entities, while the right ones include singleton entities. The upper two figures are for MUC 6 data set and the lower two are for ACE 2002 corpus.

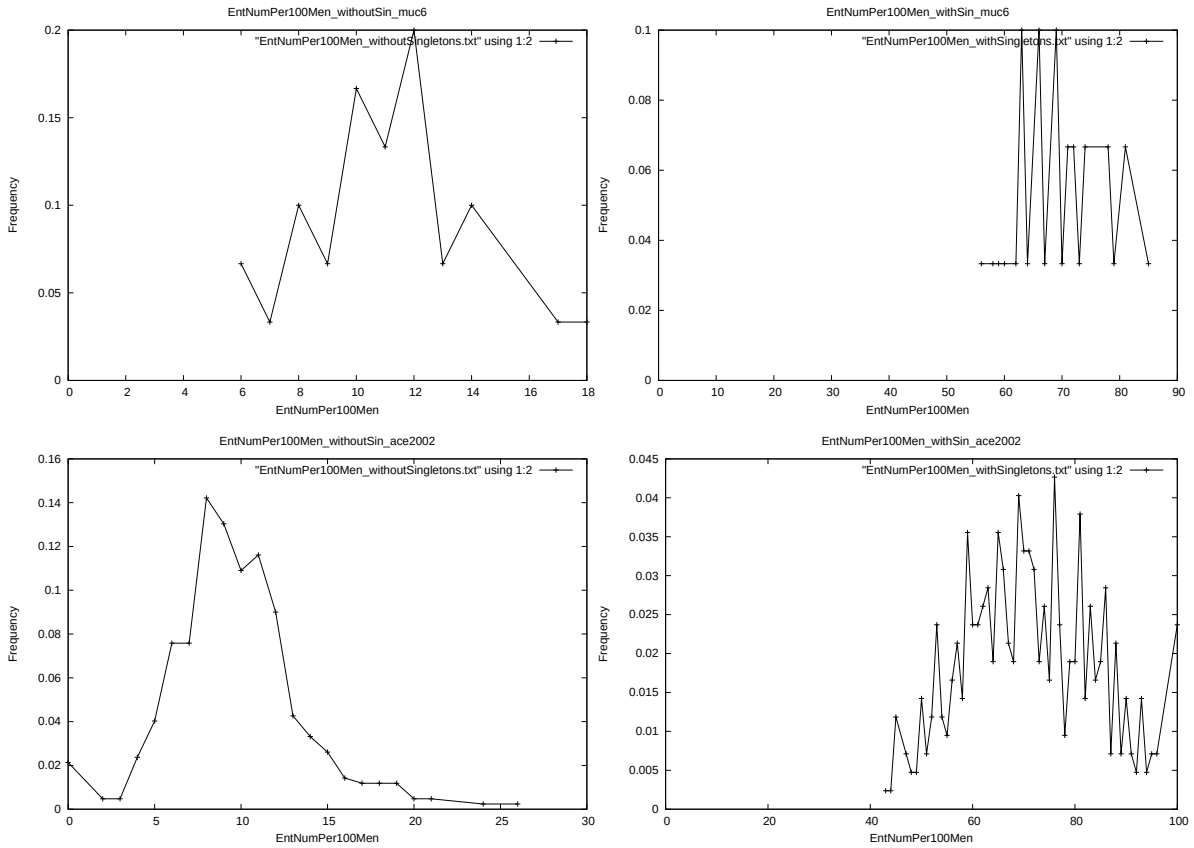

Figure 7.2: The Distributions of k With and Without Singleton Entities

It can be seen that when using system mentions (i.e. the settings with singleton entities), the distributions of the number of entities contain a lot of noise compared with the true mention setting without singletons. Such noisy distributions make the prediction of k difficult to be approached by regression methods. This motivates our proposed preference-based k *model* which does not estimate the intrinsic distribution of k, but attempts to optimize the application F-score directly.

The Performance of Our Proposed k Model. With the set of features described in Section 4.3.4, Table 7.23 gives the performance for the classification step of our proposed k model. The true and false classes correspond to the decisions which prefer the first or the second partitionings. Since the upper bound of k is decided by simply counting the numbers of different mention strings, we generate an approximately 1:6 ratio for positive and negative instances. The much bigger size of negative instances explains the low F-score the false class achieves. Although the classification performance does not directly correlate with the final coreference results, it is empirically observed that improving the classification step boosts

COPA's resolution results correspondingly.

Class	**R**	**P**	**F**
false	0.271	0.428	0.332
true	0.759	0.611	0.677

Table 7.23: k Model's Classification Performance on the CoNLL Development Data

Table 7.24 illustrates the performance of our proposed partitioning algorithms on the CoNLL development data and on the ACE 2004 development data. With the current set of the k model features, the *flatK partitioner* does not show its superiority over the *R2 partitioner*. However, it is potentially useful for incorporating global set-level information, such as the number of entities and the relations between entities. The numbers with **bestK** suggest the upper bound performance of the *flatK partitioner*. The bestK setting chooses the k's which achieve the best coreference performances.

	R2			flatK			flatK(bestK)		
	R	P	F	R	P	F	R	P	F
CoNLL									
MUC	59.99	61.82	60.89	60.04	60.99	60.51	60.51	61.97	**61.23**
B^3_{sys}	67.78	73.29	70.43	68.23	71.94	70.03	68.6	73.28	**70.86**
$CEAF_{sys}$	46.72	44.93	45.81	45.97	45.02	45.49	46.86	45.42	**46.13**
ACE04									
MUC	63.3	70.9	66.9	63.5	70.8	67.0	61.8	78.8	**69.3**
B^3_{sys}	70.9	81.0	75.6	71.0	81.0	75.7	68.8	86.2	**76.5**
$CEAF_{sys}$	71.8	67.4	69.6	71.8	67.5	69.6	71.9	69.3	**70.6**

Table 7.24: COPA R2 Vs. flatK's (with the alpha*=0.07, bold indicates significant improvement in F-score over the others according to a paired-t test with $p < 0.05$)

7.7 Summary

In this chapter, our proposed model *COPA* is evaluated in various settings. For the model comparisons, we do not include the graph partitioning algorithm proposed by Nicolae & Nicolae (2006) as a baseline system, because our adopted baseline model Bengtson & Roth (2008) is claimed to produce better performance over the previous ones. For the state-of-the-art systems after Bengtson & Roth (2008), we compare them with the CoNLL 2011 shared task setup.

***COPA* vs. Pairwise Models.** By comparing *COPA* with two pairwise models in a strict manner (i.e. leaving only the models to be different), it is suggested that the performance gains of our graph-partitioning model come from the usage of full contexts and the direct optimization of coreference sets. From the comparison experiments conducted on several corpora and with different evaluation metrics, we conclude that our global model triumphs over the pairwise methods consistently.

***COPA* vs. the State-of-the-art.** The CoNLL 2011 shared task allows us to compare our system with the state-of-the-art systems on the OntoNotes corpus, which is a big collection of documents and is well-annotated. *COPA* participates with the *R2 partitioner*, and performs competitively with only a limited amount of training documents applied (coming in as the second in the open track, and also belongs to the second block in the closed track). It is shown that *COPA* works stable on different types of documents, such as news articles and speech transcripts, and incorporating new features is simple as the learning process is very light-weighted.

***COPA*'s Domain Adaptation & Weakly Supervised *COPA*.** In order to further test the robustness of *COPA*, we also provide the experiments on a data set of clinical reports. The *flatK partitioner* is used in this setting, and the performance is encouraging that *COPA* can be easily adapted to new domains by incorporating some domain-specific knowledge.

In Section 7.5, more extensive experiments are conducted to illustrate the weakly supervised nature of the *COPA* model. Our hypergraph model is shown to be stable with respect to the amount of the training data. For the clinical set, we need as little as five percent of the training data to achieve a competitive performance. This makes *COPA* a good choice, when coreference resolution needs to be applied to new domains and new languages.

Our Proposed k *model*. We analyze our proposed k model in Section 7.6 which is designed to assist the *flatK partitioner*. We show statistics on the number of entities within documents and provide experimental numbers to show the current status of the model.

Graph models cannot deal well with positional information, such as distance between mentions or the sequential ordering of mentions in a document. We implement distance information as weights on hyperedges which results in a decent performance. However, this is limited to pairwise relations and thus does not exploit the power of the high-degree relations available in *COPA*. We expect further improvements, once we manage to include positional information directly.

An error analysis reveals that there are some cluster-level inconsistencies in the *COPA* output, such as the cluster with three mentions [*Bill Clinton*], [*Clinton*] and [*Hillary Clinton*] where [*Bill Clinton*] and [*Hillary Clinton*] are incompatible with each other. Enforcing the consistency would require a global strategy to respect the constraints during the partitioning phase. We also explore constrained clustering algorithms in *COPA*, a field which has been very active recently (Basu et al., 2009). Constrained clustering methods should allow us to make use of negative information from the cluster-level perspective (see Chapter 8 for details).

Chapter 8

The Constrained *COPA*

The Constrained *COPA*. The coreference resolution task is to cluster mentions into sets so that all mentions in one set refer to the same entity. *COPA* represents documents as hypergraphs, with relational features as hyperedges. Upon the hypergraphs, the system resorts to graph partitioning techniques to generate the final coreference sets. The partitioning should be significantly improved using supervision in the form of **pairwise constraints**, e.g. pairs of mentions which are known to be in the same coreference sets (*Must-Link* constraints) or in different ones (*Cannot-Link* constraints). The constraints suggest top-down advice to improve the output partitioning. While it is straightforward to interpret *Must-Link* constraints as highly weighted edges, there is no trivial way to include negative relations (i.e. *Cannot-Link* constraints) into a graph representation. Directly adding negative edges into a graph results in a NP-hard problem for the standard graph partitioning algorithms, although it can be addressed by specific algorithms such as correlation clustering (Bansal et al., 2002).

In this chapter, we include *Cannot-Link* constraints within the hypergraph partitioning framework of *COPA* **without changing the already-adopted spectral clustering algorithms**. The constrained *COPA* applies constrained data clustering algorithms to the vector representations in the spectral space, which are generated during the spectral clustering procedure. In this way, the consistent partitions are found by both respecting the constraints and optimizing the normalized cut. From the supervision point of view, this work of including constraints can be viewed as the first step towards **a better learning model for *COPA***. However, pairwise constraints only provide limited pairwise guidance. Improvements are expected by further exploring the learning phase of *COPA*.

Enforcing Transitivity in Coreference Resolution. In this chapter, we aim to show that including *Cannot-Link* constraints is helpful to the task of coreference resolution. In our hypergraph representation, the weight of a hyperedge indicates how close its incident vertices are to each other with respect to the corresponding relation. The vertices without edges in

between can still be clustered into the same coreference set due to the transitive closure which is implicitly done during the clustering process. Therefore, without any means to enforce the constraint respecting, inconsistent clusters can be derived. For example, when a mention [*Bill Clinton*] is connected with a mention [*Clinton*] in a graph, and at the same time a similarly weighted edge is connecting the mention [*Clinton*] and a mention [*Hillary Clinton*], the mention [*Bill Clinton*] and the mention [*Hillary Clinton*] therefore end up in the same cluster despite of the negative relation between them (e.g. different person names indicate different entities).

There have been attempts to enforce transitivity in coreference resolution, for instance, by imposing constraints on integer linear programming (ILP) (Finkel & Manning, 2008) or by disallowing inconsistent assignments during the optimization of the graphical models (the second model in McCallum & Wellner (2005)). However, we work on including constraints into graph partitioning algorithms, in order to generate more consistent coreference sets.

We experiment with both artificial clean constraints and automatically generated ones. The experiments on clean constraints show significant improvements by applying our proposed constrained partitioning algorithm. However, our experimental results with generated constraints are mostly negative, due to the low coverage of the proposed constraints. Detailed discussions on the current problem and future work are also provided.

The previous efforts on including constraints in the coreference resolution task are introduced in Section 8.1.1, and the existing general purpose constrained clustering algorithms are in Section 8.1.2. We describe our proposed algorithm in Section 8.3, and empirically analyze the performance of the constrained *COPA* in Section 8.5.

8.1 Background

8.1.1 Enforcing Transitivity in Coreference Resolution

It has been observed that the two-step coreference systems (i.e. conducting a classification step and a clustering step) tend to generate inconsistent coreference sets. Since the negative predictions from the classification step are ignored, the transitivity of the coreference relation is not enforced explicitly in the clustering step.

Constrained Clustering Methods. Cardie & Wagstaff (1999) include constraints into their distance metric to modify the edge weights between mentions, and perform graph clustering algorithms upon the modified graphs afterward. Built upon Cardie & Wagstaff's system, Wagstaff (2002) attempts to apply constrained clustering algorithms directly to the task (see her Chapter 5). To illustrate the contributions of the constraints, Wagstaff only compares

against the system that does not use constraint information at all. For instance, the *gender agreement* indicator is excluded from the feature set of the baseline system. We argue that constraints can be straightforwardly incorporated into the standard feature sets, and simply excluding constraint information leads to a very low performance of the baseline system (see column 1 of Table 5.5 in Wagstaff (2002)).

Constrained ILP Models. Klenner (2007) and Finkel & Manning (2008) impose transitivity constraints on the integer linear programming optimization (ILP) to cluster the pairwise classification decisions into sets. With constrained *COPA*, we enforce transitivity with one-step clustering algorithms. We also do not suffer from expensive computational complexity as ILP models do.

Constrained Probabilistic Models. McCallum & Wellner (2005) optimize the conditional probability of the global entity assignment, by casting the proposed graphical model as an equivalent graph partitioning problem — the correlation clustering problem (Bansal et al., 2002). Correlation clustering operates on pairwise relations between data points, to derive partitions which respect the relations as much as possible. Since negative edges are allowed in such graphs, the cluster-level consistency is taken care of directly. McCallum & Wellner use fully connected graphs with all mentions as vertices. We believe that the coreference relation can be represented in much sparser graphs as the ones adopted by *COPA* (see Chapter 4). Moreover, only a small amount of negative relations between mentions need to be considered as constraints, rather than intensively making use of many trivial ones (i.e. the negative relations between the mentions which are not likely to be clustered into the same set at all). In this thesis, we propose to guide the graph clustering algorithm to generate more consistent partitions with the selected *Cannot-Link* constraints.

Sapena et al. (2010) use a constraint-based approach (i.e. relaxation labeling) for coreference resolution with the learned constraints applied. It is shown that the proposed model outperforms an ILP algorithm which enforces transitivity constraints. The work is conceptually similar to the constrained *COPA*, except that we focus on the standard graph-clustering setup.

Entity-mention Models. Entity-mention models (Luo et al., 2004; Yang et al., 2008; Culotta et al., 2007) take care of the entity-level consistency by the incremental manner of processing. Entity-level information gets accumulated as the entities grow, the within-entity consistency is therefore maintained. Despite of the improved expressiveness, entity-mention models have not yield particularly encouraging results yet (Ng, 2010), possibly due to the seriousness of the error propagation.

8.1.2 Literature on Constrained Clustering

Due to the unsupervised nature of clustering algorithms, the obtained clusters may not necessarily be consistent with the domain knowledge of interest. For instance, in the image segmentation task, while expecting to cluster portraits of persons by gender, it is still possible to generate clusters with and without glasses in the portraits. Constrained clustering allows one to specify prior (domain) information about clusters to guide the clustering process in order to avoid creating spurious partitions.

Constrained Data Clustering. Most of the previous efforts of including constraints into clustering algorithms have been on the data which can be represented as vectors. Wagstaff & Cardie (2000) propose to modify the standard k-means algorithm (MacQueen, 1967) to make sure that no constraint is violated while assigning data points to clusters. Basu et al. (2002) use annotated data points to form k-means's initial clusters and to constrain the following assignments. Instead of modifying the assignment methods of k-means, one can also learn distance metrics from pairwise constraints (Bar-Hillel et al., 2003; Klein et al., 2002; Xing et al., 2003). Basu et al. (2004) propose a probabilistic model for semi-supervised clustering based on Hidden Markov Random Fields (HMRFs). Recently, this area has greatly expanded to include algorithms that leverage additional domain knowledge for the purpose of clustering (Basu et al., 2009).

Constrained Graph Clustering. For tasks where relations are of greater interest than data points themselves (e.g. the coreference resolution task which focuses on identifying the coreference relation) or where data vectors are not directly available, graph clustering fits more appropriately than data clustering techniques. There is only a little work on constrained graph clustering. Kamvar et al. (2003) modify the similarities between the constrained data items and then apply classifiers in the spectral space, so that spectral clustering is transformed to spectral classification. Our proposed constrained *COPA* resembles the spirit of making use of the data representation in the spectral space, but we do not apply classification steps. Kulis et al. (2005) construct appropriate kernels including constraint penalties, with which kernel k-means (Dhillon et al., 2004) can be applied to iteratively find the optimization of the corresponding objective functions. There are also attempts to combine pairwise constraints with the normalized cut directly, but only with *Must-Link* constraints (Yu & Shi, 2004) or only for two-class problems (Coleman et al., 2008).

In the constrained *COPA*, we combine a simple constrained data clustering algorithm (Wagstaff & Cardie, 2000) with our hypergraph spectral clustering algorithms (see Chapter 4) via the

spectral embedding. With our constrained clustering algorithm, we avoid modifying the constructed graphs or changing the objective functions of the original partitioning algorithms.

8.2 Inconsistency Analysis on Output Coreference Sets

Before introducing our proposal of the constrained *COPA*, we firstly provide examples of inconsistent coreference sets generated by the basic *COPA* (Chapter 4). Since we only focus on the pairwise *Cannot-Link* constraints in this chapter, the inconsistent sets are determined to be the ones containing at least one pair of mentions which do not corefer. By illustrating the spurious coreference set examples, we motivate the proposal of the constrained *COPA*.

The analysis in this section is conducted on the OntoNotes development set (see Section 3.3), and *COPA*'s *CoNLL* evaluation numbers are given in Table 8.1.

R2 partitioner	**R**	**P**	**F1**
MUC	60.87	61.92	61.39
BCUBED	68.76	72.57	70.61
CEAF(E)	46.18	45.15	45.66
overall			59.22

Table 8.1: *COPA R2 partitioner*'s results on the OntoNotes development set using *CoNLL* metrics

Frequency of Inconsistent Clusters. We collect the output coreference sets where there are at least one pair of mentions belonging to different entities. The inconsistencies are only measured between the mentions which are not twinless[1], so that their ground truth annotations are available and the effect of the mention detection is not taken into account. From Table 8.2, it can be seen that around $1/6$ of the output clusters from the basic version of *COPA* contain inconsistent mentions, occurring in half of the documents.

[1]The mentions which are not aligned with true mentions are called twinless (Stoyanov et al., 2009)

Overall Output Clusters	Inconsistent Clusters
3097 (in 202 documents)	484 (in 102 documents)

Table 8.2: Inconsistent Output Clusters from *COPA R2 partitioner* on the OntoNotes Development Set

Although there is only a small portion of the output clusters containing inconsistencies, we believe that the problem will become more severe when more relational features are included and when the graph structure becomes richer. Since the negative relations are taken as negative features in *COPA* (see Chapter 5), the violated ones in the output result from the partitioning phase only. Our objective here is to guide the partitioning algorithm with cluster-level information. It is worth noting that although the *Cannot-Link* constraints adopted in this chapter are pairwise, the consistencies are enforced on the cluster level.

Inconsistent Cluster Examples. With the inconsistent cluster examples, we aim to illustrate how they are generated via the transitivity closure automatically done during the partitioning procedure. In the examples, the subscripts of the square brackets (i.e. []) indicate the true entity assignments and the ones of the curly brackets (i.e. {}) give the system output.

In Example (1), the mention {[*He*]} is wrongly cut away from the entity JUSTICE ANTONIN SCALIA, and is grouped with the LAURANCE TRIBE entity whose name indicates female gender. This mistake is generated via the connection between the mentions {[*He*]} and {[*Tribe*]}. It shows that solely activating a negative feature between {[*He*]} and {[*Laurance Tribe, Gore's attorney*]} does not prevent this inconsistent cluster in the output. A better partitioning should be expected for this example when the cluster-level *gender agreement* constraint is respected.

Example (1):

$\{[\textit{Laurance Tribe, Gore's attorney}]_1\}_1$, said the state court did nothing illegal.

$\{[\textit{Justice Antonin Scalia}]_2\}_2$ also pressed $\{[\textit{Tribe}]_1\}_1$.

$\{[\textit{He}]_2\}_1$ said the state court relied on the Florida Constitution to draft its decision.

In Example (2), both entities ANY ECONOMIC THEORY and AN ECONOMIC THEORY are

only active locally (i.e. in their own sentences). However, they are mistakenly linked together via the definite expression {[*the theory*]}. Since it is most likely that the indefinite noun phrases introduce new entities, the connections between {[*an economic theory*]} and its preceding mentions should be forbidden. This can be easily interpreted as a *Cannot-Link* constraint.

Example (2):

For example your uncle, using $\{[\textit{any economic theory}]_1\}_1$, the probability that $\{[\textit{it}]_1\}_1$ will be accurate is virtually 0.

So whenever you discuss $\{[\textit{an economic theory}]_2\}_1$ with someone, the response would be: My uncle isn't like that, so $\{[\textit{the theory}]_2\}_1$ is baloney.

In Example (3), the mention {[*him*]} is clustered together with the mention {[*He*]}. This violates Principle B of the binding theory (see Section 2.1). When the principle is respected, the resolution of the mention {[*He*]} can be indicated by the observation that the entity RUSSIAN FOREIGN MINISTER IGOR IVANOV is more salient (i.e. in the subject position of the sentence) than the entity KOSTUNICA in this context.

Example (3):

$\{[\textit{Russian Foreign Minister Igor Ivanov}]_1\}_1$ congratulated $\{[\textit{Kostunica}]_2\}_2$ on $\{[\textit{his}]_2\}_2$ election victory .

$\{[\textit{He}]_1\}_1$ also gave $\{[\textit{him}]_2\}_1$ a letter from Russian President Vladimir Putin.

The examples introduced in this section convey that simply preventing links between non-coreferent mentions as suggested by the negative features do not ensure the within-cluster consistencies in the output. The examples also indicate that the partitioning algorithms should be improved with the guidance of linguistic knowledge. In this chapter, we focus on guidance information in the form of *Cannot-Link* constraints, and address the problem by proposing **a constrained hypergraph partitioning algorithm**.

8.3 Our Proposal — the Constrained *COPA*

In this section, we propose to combine constrained data clustering algorithms with our hypergraph spectral clustering algorithms via the spectral embedding. The proposed method avoids changing the objective function of the adopted hypergraph clustering algorithms. It also avoids propagating the constraints on the originally constructed hypergraphs. Our proposal makes it feasible to apply different constrained data clustering algorithms within the spectral graph clustering framework.

A simple constrained data clustering algorithm *COP-KMeans* is introduced in Section 8.3.1, and our variant of the *COP-KMeans* is in Section 8.3.2. Section 8.3.3 describes our proposal of combining the modified *COP-KMeans* with *COPA* via the spectral embedding, in order to tackle the constrained hypergraph clustering problem.

8.3.1 Constrained Data Clustering — *COP-KMeans*

The standard k-means algorithm (MacQueen, 1967) iteratively assigns data points to their closest clusters, and converges when there are no more changes in the cluster assignments. The k-means algorithm solely depends on the intrinsic distributions of the given data sets. Wagstaff & Cardie (2000) provide a modified version of the k-means algorithm which makes use of the background knowledge being expressed as pairwise constraints. Their proposed variant *COP-KMeans* respects the pairwise constraints during the cluster assigning process. The algorithm disallows the assignments where constraints are violated, therefore resulting in consistent partitions. There are two types of pairwise constraints which are prevalently adopted and are the input to *COP-KMeans*.

- A **Must-Link** constraint suggests that the given pair of data points should belong to the same cluster.
- A **Cannot-Link** constraint suggests that the given pair of data points should not belong to the same cluster.

Algorithm 6 gives the details on *COP-KMeans*. Line 4 and Line 5 of the algorithm locate the modifications *COP-KMeans* makes upon the standard k-means algorithm. Instead of assigning a data point to the closest cluster, *COP-KMeans* checks on the constraint violation first. Only the clusters which do not violate any given constraints are considered in the assignment.

Algorithm 6 *COP-KMeans* Algorithm (single iteration) (Wagstaff & Cardie, 2000)

1: **input**: data set D, must-link constraints $Con_{=} \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$
2: Let $C_1 \ldots C_k$ be the initial cluster centers
3: **for** each point d_i in D **do**
4: Assign d_i to the closest cluster C_j such that $violateConstraints(d_i, C_j, Con_{=}, Con_{\neq})$ is false
5: If no such cluster exists, fail (**return** $\emptyset$)
6: **end for**
7: **for** each cluster C_i **do**
8: Update the center of C_i by averaging all of the points d_j that are assigned to C_i
9: **end for**
10: **return** partitioned $C_1 \ldots C_k$

The $ViolateConstraints$ function in Algorithm 7 suggests that the pairwise constraints are brutally enforced in *COP-KMeans*. No partitioning output is generated when there is no single assignment respecting all given constraints (i.e. Line 12).

Algorithm 7 $ViolateConstraints$ Function Algorithm (Wagstaff & Cardie, 2000)

1: **input**: data point d, cluster C, must-link constraints $Con_{=} \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$
2: **for** each $(d, d_{=}) \in Con_{=}$ **do**
3: **if** $d_{=} \notin Con_{=}$ **then**
4: **return true**
5: **end if**
6: **end for**
7: **for** each $(d, d_{\neq}) \in Con_{\neq}$ **do**
8: **if** $d_{\neq} \in Con_{\neq}$ **then**
9: **return true**
10: **end if**
11: **end for**
12: **return false**

8.3.2 Our Variant of *COP-KMeans*

Since *COPA* is an end-to-end system which works in a noisy environment, enforcing constraints in a hard way as *COP-KMeans* does can be problematic. We propose a variant of *COP-KMeans* to minimize the number of the violated constraints. The proposed *VD-KMeans* is given in Algorithm 8, with the modification in line 4 replacing the $ViolateConstraints$ function with the $ViolationDegree$ function (see Algorithm 9). $ViolationDegree$ counts the number of the violated *Cannot-Link* constraints when assigning a data point to a cluster, and *VD-KMeans* simply decides on the cluster with the smallest violation degree or on the closest cluster when the violation degrees are the same.

Algorithm 8 *VD-KMeans* Algorithm (single iteration)

1: **input**: data set D, cannot-link constraints $Con_{\neq} \subseteq D \times D$
2: Let $C_1 \ldots C_k$ be the initial cluster centers
3: **for** each point d_i in D **do**
4: Assign d_i to the cluster C_j with the smallest $ViolationDegree(d_i, C_j, Con_{\neq})$
5: For clusters are with the same violation degree, choose the closest one
6: **end for**
7: **for** each cluster C_i **do**
8: Update the center of C_i by averaging all of the points d_j that are assigned to C_i
9: **end for**
10: **return** partitioned $C_1 \ldots C_k$

Algorithm 9 $ViolationDegree$ Function Algorithm

1: **input**: data point d, cluster C, cannot-link constraints $Con_{\neq} \subseteq D \times D$
2: **for** each $(d, d_{\neq}) \in Con_{\neq}$ **do**
3: **if** $d_{\neq} \in C$ **then**
4: Increase the violation degree: $vdCnt++$
5: **end if**
6: **end for**
7: **return** $vdCnt$

We only consider *Cannot-Link* constraints in constrained *COPA*, as *Must-Link* constraints can be straightforwardly incorporated as highly weighted hyperedges in our hypergraph models.

8.3.3 Constrained Hypergraph Spectral Clustering

The hypergraph-based spectral clustering has been introduced in Section 4.2.2. In short, spectral clustering reduces the data dimensionality by using the eigenvectors of the graph Laplacians. The resulting vector representation of the data set is the *spectral embedding*. For the sake of the expressive convenience, we start with revisiting some of the basic notations.

Hypergraph Normalized Cut. When the normalized cut (*Ncut* (Shi & Malik, 2000)) is adapted to hypergraphs (Zhou et al., 2007), it preserves the intuition that a good partitioning cuts as few hyperedges as possible while leaving the resulting partitions as dense as possible. The hypergraph *Ncut* for a k-partitioning P_k is defined by

$$Ncut(P_k) = \sum_{1 \leq i \leq k} \frac{vol\partial V_i}{volV_i} \tag{8.1}$$

$P_k = \{V_i | V = V_1 \cup V_2 \cup \cdots \cup V_k\}$, where $V_i \cap V_j = \emptyset$, for all $1 \leq i, j \leq k$ and $i \neq j$. The volume $volV_i$ gives the within-cluster density of the the vertex set V_i. The volume of the hyperedge boundary ∂V_i measures the hyperedges to be cut in order to derive V_i as a cluster. The objective of our partitioning algorithm is therefore to minimize Equation 8.1.

The Spectral Embedding. The *Ncut* value can be minimized using a relaxation approach, which approximates discrete cluster memberships with continuous real numbers. The approximation can be approached by solving the eigen problem of the *hypergraph Laplacian*:

$$L = I - {D_v}^{-\frac{1}{2}} H W {D_e}^{-1} H^T {D_v}^{-\frac{1}{2}} \tag{8.2}$$

Let $(\lambda_i, v_i), i = 1, \ldots, n$, be the eigenvalues and the associated eigenvectors of L, where $0 \leq \lambda_1 \leq \cdots \leq \lambda_n$ and $\|v_i\| = 1$. The continuous solution to the *Ncut* minimization is then provided by a new low-dimensional data representation X:

$$X = (v_1, \cdots, v_k) \tag{8.3}$$

where X is called the k-th order *spectral embedding* of the graph. It has been shown that k generally equals to the number of clusters (Ng et al. 2001). A standard data clustering algorithm, such as the k-means, can be applied to cluster the graph nodes in the new space afterward.

Applying Constrained Data Clustering Algorithms to the Spectral Embedding. Figure 8.1 illustrates our proposal of the constrained spectral graph clustering algorithm. The *Cannot-Link* constraints are extracted from the graph to be partitioned, and are imposed on the generated spectral embedding. Since the spectral embedding transforms the original graph

to a vector representation of vertices, constrained data clustering algorithms can be directly applied.

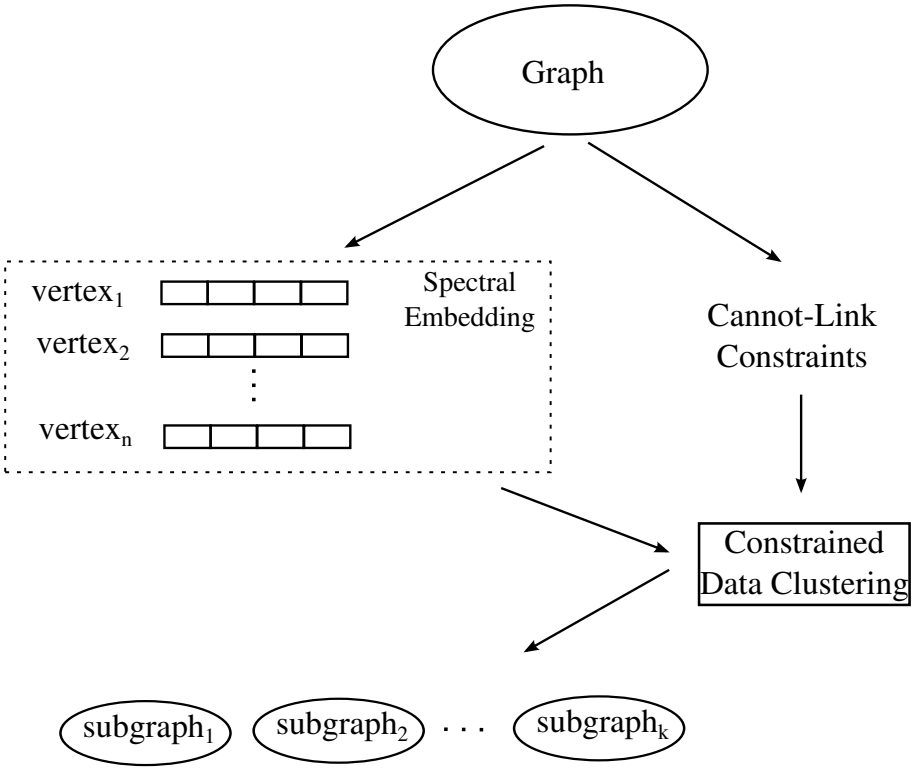

Figure 8.1: Illustration of Constrained Spectral Graph Clustering

8.3.4 Constrained *COPA* Partitioners

COPA implements a hierarchical multi-class partitioner, *R2 partitioner*, which recursively bi-partitions the hypergraph until a stopping criterion (i.e. α^*) is reached (see Section 4.3.3.1). We propose to apply constraints to each recursion of the *R2 partitioner*. The resulting ***ConR2 partitioner*** is outlined in Algorithm 10. *ConR2 partitioner* recursively bi-partitions when the *Ncut* value is smaller than α^* or when the violated constraints are fewer compared with the input hypergraph (i.e. Line 8). The current bi-partition is not accepted when the constraint violations do not become fewer after partitioning (i.e. Line 11). *VD-KMeans* is used as the data clustering algorithm, taking the spectral embedding and *Cannot-Link* constraints as input.

Algorithm 10 *ConR2 partitioner*

input: target hypergraph HG, *Cannot-Link* constraints CN, $\alpha^\star$
Counts the violated constraints $VioCnt$ for the input HG
Solve for the 2-nd spectral embedding, SE
Generate two sub HG's using *VD-kmeans*(SE, CN)
Counts the violated constraints $VioCnt_1$, $VioCnt_2$ for two sub HG's
if $\min_i(Ncut_i) < \alpha^*$ OR both $VioCnt_i$'s are smaller than $VioCnt$ **then**
  for each sub HG **do**
    Bi-partition the sub HG with *R2 partitioner*
  end for
else
  if any $VioCnt_i$ is bigger than or equal to $VioCnt$ **then**
    Output the input HG
  end if
else
  Output the current sub HG
end if
output: partitioned HG

The *R2 partitioner* optimizes the bi-partition at each recursion step. However, it is not guaranteed that the final output clusters are globally optimized due to the hierarchical nature. To overcome the problem, we experiment with the *flatK partitioner* (see Algorithm 2) as well [2]. However, the *ConflatK partitioner* is not covered in this chapter.

8.4 *Cannot-Link* Constraints for Coreference Resolution

The Difference Between Negative Features and Cannot-Link Constraints. In this section, we describe the *Cannot-Link* constraints proposed for coreference resolution. The *Cannot-Link* constraints are negative relations between a pair of mentions, and are at the same time taken as negative features too. Negative features in *COPA* prevent hyperedges to be built during the graph construction phase, while the *Cannot-Link* constraints guide the partitioners in

[2]With the constrained *ConflatK partitioner*, k clusters are output simultaneously. The *VD-kmeans* algorithm is again applied to the k-th spectral embedding of the input hypergraph, and directly outputs the final clusters. The model used to predict the k is introduced in Section 4.3.4.

the inference procedure. Duplicating the constraints as negative features enables us to analyze the contributions which are solely from the constrained clustering algorithm.

(1) CN_Gender

- Two mentions do not agree in gender.
- For instance, the mentions [*Hillary Clinton*] and [*he*] should not be clustered into one set due to the incompatible gender.

(2) CN_ContraMod

- Two mentions have the same syntactic heads, and the anaphor has a modifier which does not occur in the antecedent or which contradicts the modifiers of the antecedent.
- For instance, a *Cannot-Link* constraint is built between [*1,000 coal rail cars*] and [*the 1,450 coal rail cars*], as the two mentions contain different quantitative modifiers.

(3) CN_ContraGPE

- Two mentions realizing different GPEs should not be in one set.
- For instance, a negative relation exists between the mentions [*Syria*] and [*Lebanon*] because they are different countries. A gazetteer consisting of lists of country names and city names is looked up for computing this constraint.

(4) CN_ContraSubjObj

- Two mentions are in the subject and object positions of a non-copular verb, and the anaphor is not a possessive pronoun.
- Considering the text "[*John*] talks to [*him*]", where the mention [*John*] should not be coreferent with the pronoun [*him*]. The dependency tree is used to identify the verbs on which the mentions depend. This constraint is derived from Principle B of the Binding theory (Section 2.1.2).

(5) CN_Span

- A mention spanning another one cannot be linked to it, except for *RoleAppositive* cases.
- Considering the embedding mentions [[*his*] *brother*], the two should not be clustered together.

(6) CN_ContraPerson

- Two person mentions with different names cannot be linked.
- For instance, the mention [*Mr. Wright*] should not be coreferent with the mention [*Mr. Valenti*] due to the different family names of the two person entities.

The Cleanness of the Proposed Constraints. Table 8.3 analyzes the cleanness of the proposed constraints. The statistics corresponds to the frequencies of the constraints holding on the OntoNotes training data. The negative signs in the table indicate that the *Cannot-Link* constraints are negative relations between mentions.

Constraints	Statistics
(1) CN_Gender	-0.993
(2) CN_ContraMod	-0.980
(3) CN_ContraGPE	-0.992
(4) CN_ContraSubjObj	-0.997
(5) CN_Span	-0.996
(6) CN_ContraPerson	-0.961

Table 8.3: The Cleanness of the Cannot-Link Constraints on the OntoNotes Training Set

8.5 Experiments on the Constrained *COPA*

Experimental Settings. In this section, we experiment with the proposed constrained *COPA*. The numbers are reported on the OntoNotes development set, using the unweighted average of *MUC*, *BCUBED* and *CEAF(E)* (i.e. the final score in CoNLL 2011 shared task). The setting of *COPA* using the *R2 partitioner* is denoted as ***R2***, upon which the setting ***R2+N_Feats*** includes the *Cannot-Link* constraints as negative features. The baseline system ***PostR2*** encodes the standard k-means algorithm and keeps bi-partitioning until there is no violated constraint any more. ***ConR2*** corresponds to the constrained *COPA* proposed in this chapter.

In Section 8.5.1, we first experiment with the clean constraints which are generated from the ground truth annotations. Such upperbound setting allows us to evaluate the proposed method while excluding the effect of the constraint generation phase. The automatically generated constraints are tested in Section 8.5.2, where the constrained *COPA* performs in a fully automatic manner.

8.5.1 Experiments with Artificial Clean Constraints

The Generation of Clean Constraints. The clean constraints are only generated for the mentions which can align with the true mentions. In this way noise brought by the twinless mentions is still kept, otherwise building clean constraints for all mentions will directly remove the spurious ones. There are a total of $144,858$ clean constraints generated for the OntoNotes development set.

***ConR2* vs. Baselines.** Table 8.4 gives the performance of our proposed constrained *COPA* with clean constraints. The difference between *PostR2* and *ConR2* is that *PostR2* only uses the constraints as the stopping criterion for the recursive partitioning, but *ConR2* actually guides the partitioning inference with the constraints.

	R2			*R2+N_Feats*			*PostR2*			*ConR2*		
	R	P	F	R	P	F	R	P	F	R	P	F
MUC	60.85	61.93	61.39	61.81	64.06	62.92	59.6	64.67	62.03	62.66	67.6	**65.03**
BCUBED	68.68	72.59	70.58	69.6	76.28	72.78	67.62	78.58	72.69	69.8	80.0	**74.55**
CEAF(E)	46.19	45.13	45.66	47.85	45.72	46.76	49.47	44.8	47.02	50.03	45.49	**47.65**
overall			59.21			60.82			60.58			**62.41**

Table 8.4: *ConR2* vs. Baselines with Clean Constraints on the OntoNotes Development Set (bold indicates significant improvement in F-score over *PostR2* according to a paired-t test with $p < 0.05$)

The improvement *ConR2* achieves compared with the setting *R2+N_Feats* demonstrates the contribution which is solely from the proposed algorithm. The precision of all metrics (except for *CEAF(E)*) are improved by using the constrained clustering algorithm. This is not surprising given the fact that *Cannot-Link* constraints are applied to prevent spurious linkages. Gains on recall are observed too. Since constraints participate in the partitioning decisions when using *ConR2*, the recall improvements suggest that the corrections on some mentions (which are involved in the constraints) also improve the resolutions of others.

The baseline system *PostR2* greedily partitions the clusters which violate constraints, without incorporating constraint information into the partitioning decisions. The *PostR2* results also produce higher precision (except for the *CEAF(E)* metric), but suffer from a bigger loss in recall. This confirms again that the constraints need to be enforced on the cluster level during the partitioning inference.

***ConR2* with Randomly Sampled Constraints.** Figure 8.2 plots the performance curve of *ConR2* given the increasing number of *Cannot-Link* constraints. The used constraints here are randomly sampled from the full set of clean constraints as introduced previously. It is worth noting that all the original clean constraints are included as negative features throughout the experiments, and only the ones used as *Cannot-Link* constraints differ in size. Therefore the leftmost points in all three plots correspond to the performance of the *R2+N_Feats* model.

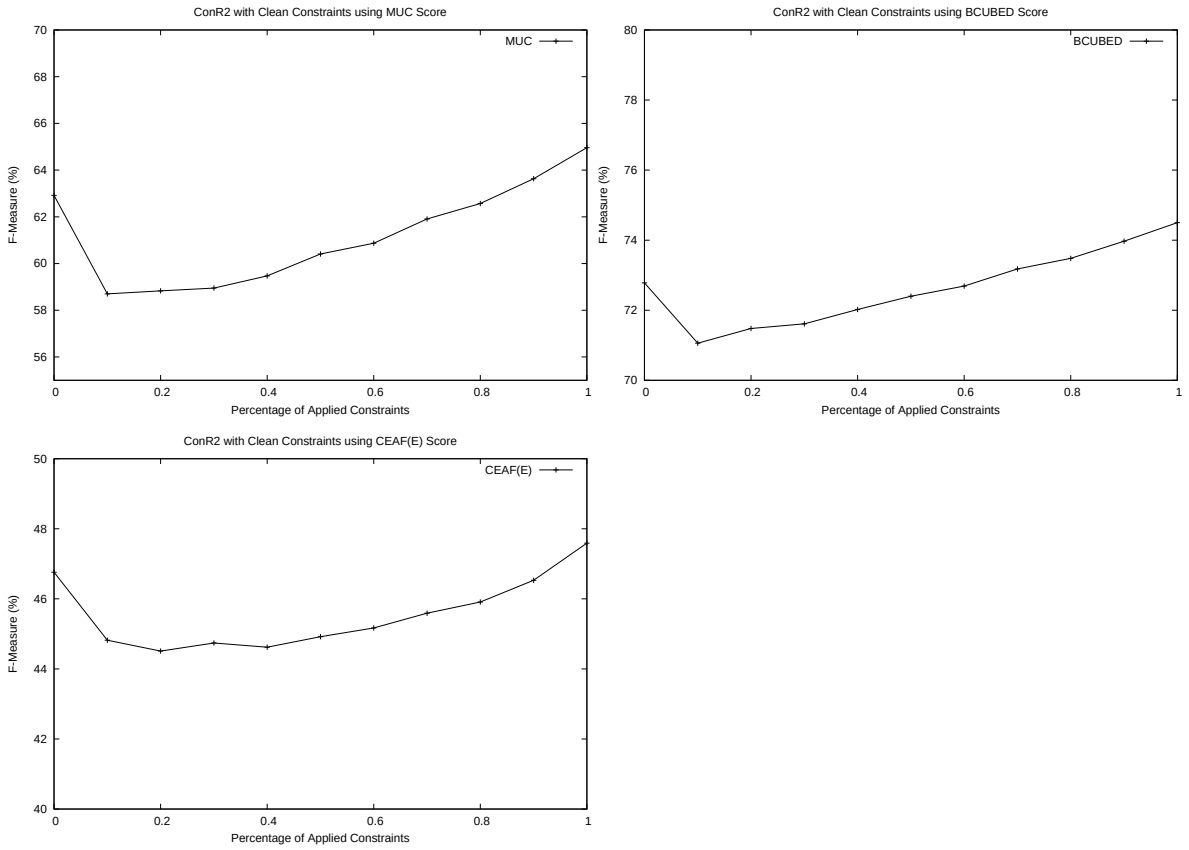

Figure 8.2: *ConR2* Performance with Increasing Size of Clean Constraints

Figure 8.2 shows that *ConR2* only outperforms *R2+N_Feats* when more than 80% of the constraints (around $115,880$) are used. Smaller sets of constraints generate worse performance compared with the *R2+N_Feats* system which does not use constraints at all. The possible explanation is that more constraints help to generate balanced clusters, while a few can easily skew the *ConR2* partitioner. This demonstrates a drawback of the proposed algorithm, that enforcing the constraints is a higher priority than deriving a good partitioning.

We conduct another group of experiments by adding noise constraints, as shown in Figure 8.3. Noise constraints are randomly sampled, and are added upon the full set of the clean constraints. The straight lines in all plots indicate the performance of the baseline *R2+N_Feats*. *ConR2*'s performance drops below the baseline soon after about 10% noise constraints are included, and keeps decreasing quickly.

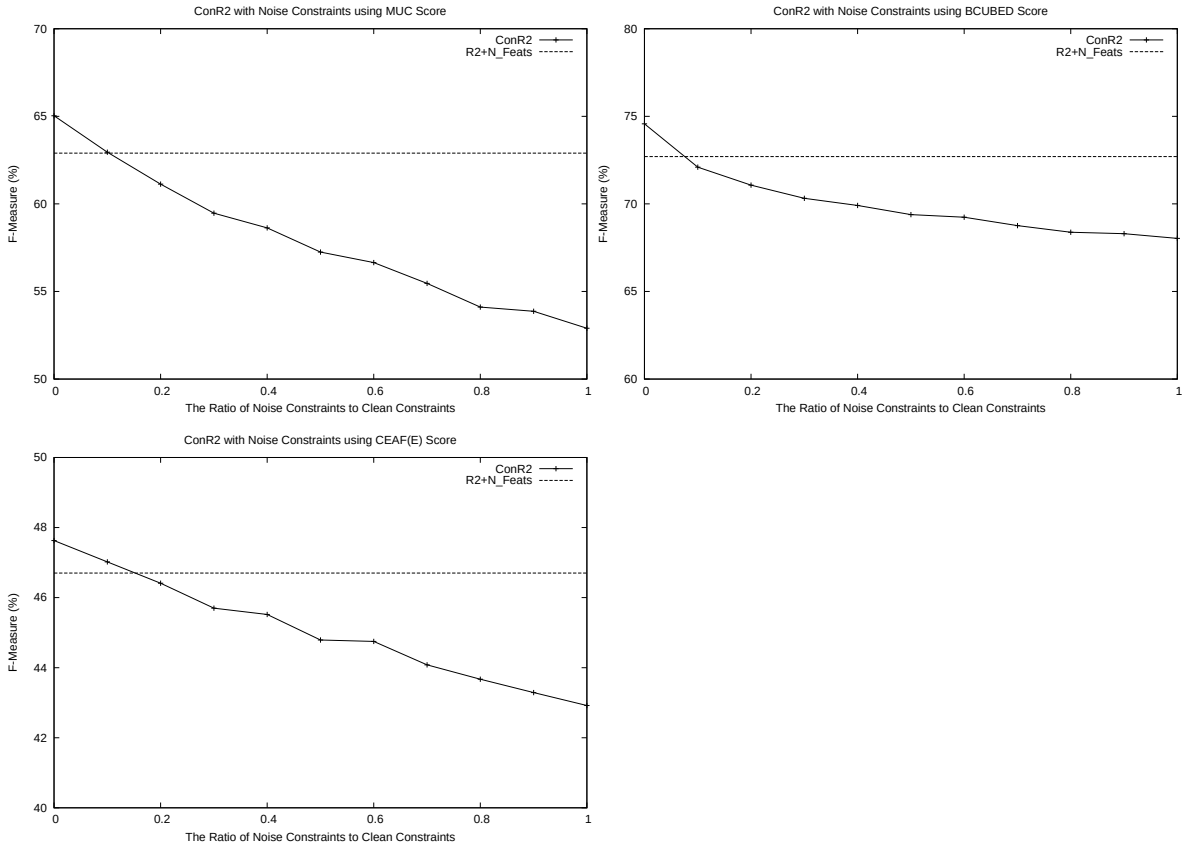

Figure 8.3: *ConR2* Performance with the Increasing Size of Noise Constraints

In this section, we experiment with the artificial *Cannot-Link* constraints using the proposed constrained *COPA*. We analyze the influence of the size of applied constraints and the size of the involved noise constraints (i.e. incorrect constraints). Significant improvement is achieved when a big enough set of constraints is provided and when the set consists of less than 10% spurious ones. The experiments on the randomly sampled clean constraints suggest a reasonable **recall** range for designing the real constraints, and the experiments on the noise constraints hint on a proper **precision** range. In the following section, experiments with the automatically generated constraints (i.e. the real constraints) are provided.

8.5.2 Experiments with Automatically Generated Constraints

***ConR2* vs. *R2+N_Feats*.** Table 8.5 shows the results of *ConR2* using the *Cannot-Link* constraints proposed in Section 8.4. Since the constraints are already included as negative features in the basic *COPA*, the *R2* performance in Table 8.4 is the same as the baseline performance in Table 8.5 (i.e. *R2+N_Feats*).

	R2+N_Feats			*ConR2*		
	R	P	F	R	P	F
MUC	60.85	61.93	61.39	59.58	61.77	60.66
BCUBED	68.68	72.59	70.58	67.57	73.22	70.28
CEAF(E)	46.19	45.13	45.66	46.6	44.47	45.51
overall			59.21			58.82

Table 8.5: *ConR2* vs. *R2+N_Feats* with Automatically Generated Constraints on the OntoNotes Development Set

From the statistics provided in Table 8.3, it can be seen that more than 90% of our automatically generated constraints are correct. This is demonstrated in the previous section to be a good proportion in order to improve upon the *R2+N_Feats*. However, *ConR2* yields worse results compared with the baseline system. It can be partially explained by the small size of the applied constraints, which is $12,555$ for the entire development set. The contributions of the proposed constraints are illustrated in Table 8.6, ordered in accordance with the cleanness of the constraints. Increases in precisions are observed for both *MUC* and *BCUBED*, but a bigger loss loss in recalls constantly occurs.

The current constrained *COPA* unfortunately generates negative results. A detailed inspection shows that several inconsistent output clusters (see Table 8.2) are not covered by the proposed constraints. For instance, (2) CN_ContraMod does not capture the negative relation between the mentions [*China's Red Cross Society*] and [*the international Red Cross Organization*]. Since the current constraints target at high precisions, more high-recall ones should be developed.

	MUC			*BCUBED*			*CEAF(E)*		
	R	P	F	R	P	F	R	P	F
(4) CN_ContraSubjObj	60.22	61.73	60.97	68.19	72.8	70.42	46.26	44.79	45.51
+ (5) CN_Span	60.22	61.78	60.99	68.15	72.86	70.42	46.33	44.81	45.56
+ (1) CN_Gender	59.93	61.76	60.83	67.87	73.01	70.35	46.42	44.63	45.51
+ (3) CN_ContraGPE	59.85	61.7	60.76	67.78	72.95	70.27	46.44	44.63	45.52
+ (2) CN_ContraMod	59.69	61.74	60.7	67.68	73.1	70.28	46.53	44.53	45.51
+ (6) CN_ContraPerson	59.58	61.77	60.66	67.57	73.22	70.28	46.6	44.47	45.51

Table 8.6: The Contributions of the Proposed *Cannot-Link* Constraints

Solved Example by *ConR2*. Although *ConR2* does not generate promising results yet, we now show an example which is solved by applying the constrained clustering algorithm. Figure 8.4 shows the output clusters by the basic version of *COPA*, where the entity PRESIDENT SLOBODAN MILOSEVIC is mistakenly mixed with the entity PRESIDENT PUTIN. This happens because both persons are male presidents and they are linked together via other mentions such as [*the president*] and [*he*].

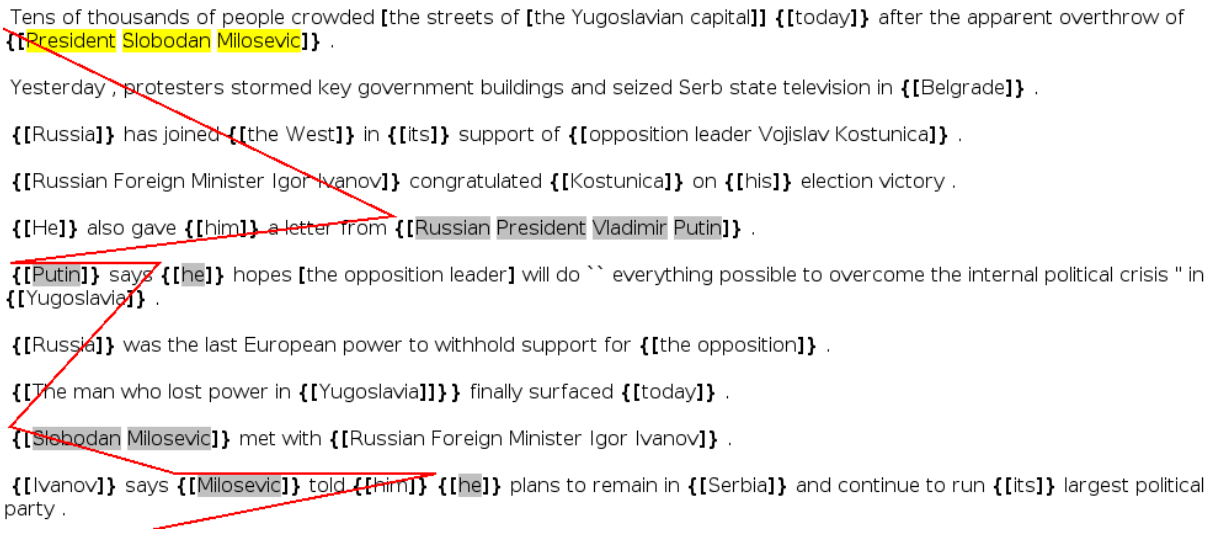

Figure 8.4: Example Output Clusters Using the Basic *COPA*

By applying the constrained *COPA*, it can be seen from Figure 8.5 that the two entities are correctly resolved thanks to the constraint (6) CN_ContraPerson.

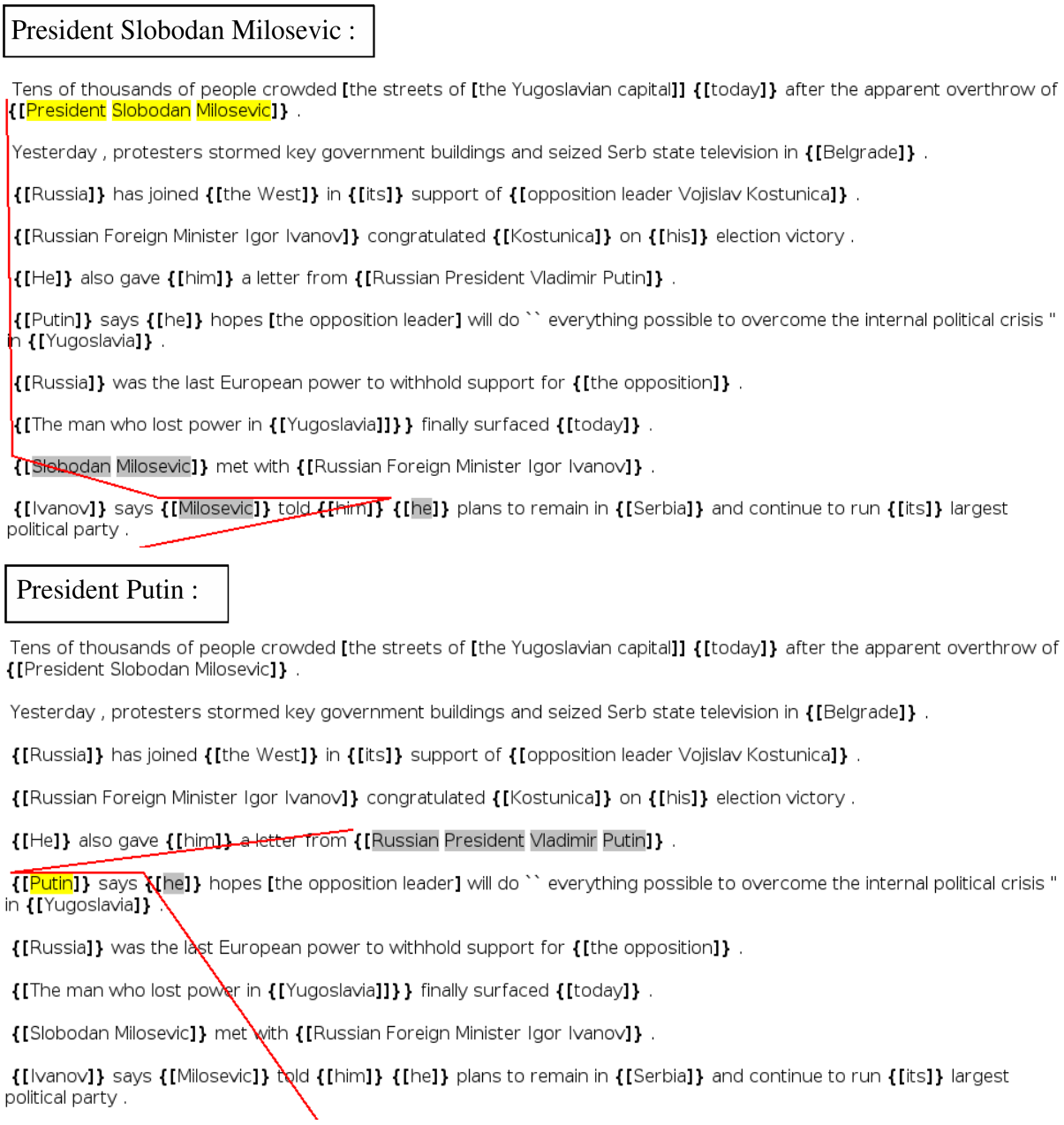

Figure 8.5: Example Output Clusters Using the Constrained *COPA*

8.6 Summary

Incorporating Constraints into Coreference Resolution. In this chapter, we consider a general problem for the clustering field. Due to the transitive closure which is implicitly done during the clustering phase, counter-intuitive clusters can be derived. This is also an

issue for the coreference resolution task when the coreference sets are generated by clustering models. For instance, the mention [*a Norwegian Transport Ship*] is clustered together with a preceding mention [*The damaged ship*] via another mention [*the ship*] which appears later in the document. However, the indefinite article "a" strongly indicates that the mention [*a Norwegian Transport Ship*] is not anaphoric. Such information can be interpreted as pairwise constraints: *Must-Link* asks the mentions to be in one cluster and *Cannot-Link* forbids so.

In order to generate consistent coreference sets, there has been previous work on enforcing transitivity for coreference resolution (e.g. Finkel & Manning (2008)) and on applying correlation clustering to incorporate negative edges in graphs (e.g. McCallum & Wellner (2005)). In this thesis, we focus on incorporating the pairwise constraints within the graph spectral clustering framework.

Our Proposal: Constrained *COPA*. In this chapter, we extend the basic version of *COPA* in order to guide the partitioning algorithms with pairwise constraints. Since the *Must-Link* constraints can be straightforwardly included as strong edges in a graph model, we only deal with *Cannot-Link*'s for now. We propose to combine constrained data clustering algorithms with hypergraph spectral clustering algorithms via the spectral embedding. In this way, we address the constrained graph clustering problem without changing the clustering objective function or modifying the originally constructed graph structures.

We conduct experiments with the constrained *COPA* on both the artificial clean constraints and the automatically generated ones.The experiments on clean constraints allow us to study the effect of the size of constraints and the proportion of the noise on the proposed algorithm. Although the improvement achieved by using the clean constraints is significant, our results on the automatically generated ones are unfortunately negative. The possible reason is that the current *Cannot-Link* constraints do not have enough coverage on the data set. Testing with constraints of a small coverage does not convey the effectiveness of the algorithm, especially when the number of the inconsistent clusters to be solved is not very big in the first place.

Future Work. Since the number of the inconsistent clusters will grow bigger when the graph structures become richer, the importance of providing prior information to guide the clustering algorithms remains. Our proposed method provides a way to address the problem with relatively little effort on adapting the original clustering algorithms. The next step for us is to include more constraints in order to explore the potential of the constrained *COPA*. We currently exclude the negative relations such as *semantic class agreement* and *number agreement*, to avoid too much noise. However, the experiments with clean constraints suggest that at most 10% noise is allowed, which is the case for both of them. So it will be reasonable to include more high-recall constraints in the future.

Chapter 9

Conclusions

Natural Language Processing (NLP) tasks process texts automatically on the syntactic, semantic and pragmatic levels, targeting at the full text understanding. Coreference resolution has been one of the most fundamental NLP task for decades, which links the referring expressions of the same entities into sets. From a pragmatic point of view, a text can be considered as a collection of entities and the relations between them. Resolving the referring expressions therefore enables us to identify the entities in a document. Furthermore, the local context of the different occurrences of an entity are implicitly merged via the coreference relation built between the referring expressions. Therefore it is made easier to extract the relations between entities from their enlarged context.

In the introduction of this thesis, we interpret the coreference relation as a high-dimensional relation, which can be derived from multiple basic relations (e.g. string similarity and semantic relatedness). Unlike the previous methods which collapse the basic relations before the inference step, we aim to maintain the basic relations until the final inference procedure. In order to do so, we propose **a hypergraph model to represent a document** as shown in Figure 9.1 (a).

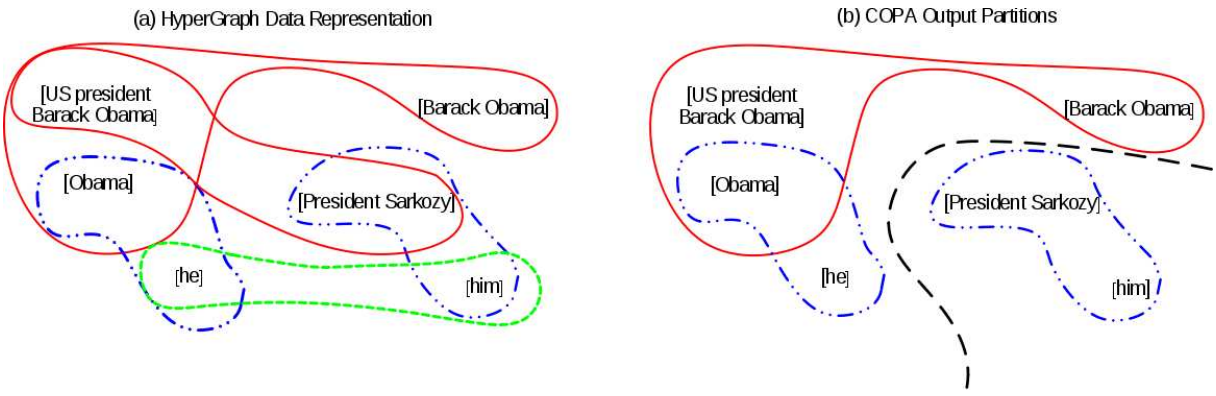

Figure 9.1: COPA Example: Processing Illustration

The thesis presents our proposed coreference system ***COPA*, an end-to-end hypergraph-partitioning-based model**. Upon the hypergraph representation of documents, partitioning algorithms are proposed to derive the coreference sets as shown in Figure 9.1 (b). By making use of the graph partitioning technique, *COPA* is able to generate the coreference sets at one step by considering all the relations encoded in the hypergraph together. In contrast to the local coreference models, our system performs the inference procedure in a global manner; and unlike the probabilistic global methods, our partitioning algorithms do not involve sophisticated probability estimations but achieves more competitive performance.

In this chapter we summarize the main contributions of our work and point out the possible future research directions.

9.1 Main Contributions

In this thesis, we address four important questions concerning the coreference resolution modeling and the end-to-end coreference system designing.

Representing the High-dimensional Coreference Relation. *COPA* represents the mentions as vertices in the **hypergraph model**, and connects them with weighted hyperedges which are directly derived from the basic relations (i.e. features). Since this allows for multiple hyperedges existing between mentions, the basic relations are incorporated into the hypergraphs in an overlapping manner. The hypergraph provides us with a way to make the coreference decisions only during the inference phase, in contrast to the previous work which combines the basic relations into the coreference relation during the graph construction phase (i.e. the representing phase).

We propose to categorize the coreference features into three types. The negative features prevent the hyperedges to be built between mentions, indicating the non-coreferential relations. The positive features are used to construct the hypergraphs, which are mainly the strong indicators for the coreference relation. The weak features enrich the hypergraph structures by providing many weak hyperedges which do not strongly correlate with the coreference relation but are still informative. The **feature categorization** is important for applying graph models in end-to-end systems, making them less sensitive to the noise and making it easier to incorporate more features.

Inferring the Coreference Sets Globally. The coreference resolution task is to derive the coreference sets from a collection of mentions. We argue that the coreference models should not only analyze the relations between mentions but also consider the relations between different coreference sets. The hypergraph partitioning algorithms adopted in *COPA* manage to

optimize the output coreference sets directly instead of only making the best decisions for mention pairs. Moreover, in our model resolving one mention depends on the resolutions of all the others, which makes *COPA* a global method.

In this thesis, we also explored **a constrained version of *COPA***. We demonstrate the importance of enforcing the transitivity in the coreference resolution task and propose to address the problem within the constrained graph clustering framework. The idea of our method is to combine the constrained data clustering algorithms with the spectral graph clustering ones via the spectral embedding. Due to the low coverage of the automatically generated constraints, our experimental results are mostly negative so far. However, the clean (artificial) constraints show promising improvements from the proposed algorithm. We leave the work on incorporating the generated constraints in *COPA* as a future research direction.

Evaluating the End-to-end Coreference Systems. In this thesis, we report the problems of the existing coreference evaluation metrics when they are applied to end-to-end system output. In order to evaluate the coreference task in a realistic setting, we propose **two variants of the evaluation metrics** B^3 and *CEAF* . Our variants are empirically shown to evaluate the noisy coreference output in an adequate way. The appropriate evaluation metrics are essential especially when the coreference systems optimize with respect to the final coreference output.

Learning Cheaply. Due to the overlapping manner of the hyperedges, *COPA* only needs to learn the weights for the basic relations instead of a high-dimensional combination of them. It requires only a few training documents to collect the simple statistics for the weights of the basic relations, so *COPA* is considered as a **weakly supervised** system. The experiments also confirm that *COPA* achieves competitive results with a small training set. This makes *COPA* a good candidate when moving to a different domain or a different language where not enough ground truth annotation is available.

9.2 Future Work

In this section, we highlight a couple of possible future research directions which should be worth investigating.

More Coreference Features. Due to the well-defined hypergraph representation and the feature categorization strategy in *COPA*, it requires little effort to incorporate relational features. The current version of *COPA* only adopts a standard set of coreference features, and it should be further improved by designing more linguistic- and world- knowledge. For instance, weak features enable us to include (a large amount of) noisy relations extracted from

the Internet such as word associations.

Besides building relations between mentions, it will be also interesting to explore the relations between mention contexts. For instance, the mentions participating in the same event as the same roles or having the same relations with the same (another) entity should have a good possibility to be coreferent with each other.

In brief, more features will help to generate hypergraphs with richer structures, and therefore better partitions should be produced on such hypergraphs.

Learning to partition. The learning scheme currently adopted by *COPA* is only to collect simple statistics about the basic relations. The constrained *COPA* can be viewed as a first step towards a better learning of our hypergraph-partitioning-based model. However, it should be worth efforts to find a learning algorithm which can directly optimize the hyperedge weights with respect to the partitioning criterion (i.e. the *NCut* value). In general, the learning procedure being consistent with the inference procedure should be able to make the most of the training data.

Graph-partitioning-based Entity Model. Although the hyperedges in *COPA* are able to represent sets of multiple mentions, we have not yet modeled entities explicitly . Enabling properties on hyperedges may be able to capture entity-level information, and such information can be propagated to mentions and vice versa via the edge-vertex incidences.

Incrementally or iteratively partitioning the hypergraphs can be another way to model entities. Entities derived from the previous runs or iterations should help with later partitionings.

Application to Other Languages and Domains. *COPA* has been lately tested on different languages, such as Chinese. It performed stable by borrowing some of the language-independent features from the English implementation, such as *head match*. As discussed in the thesis already, the proposed system performs competitively across different domains too. In the future, it will be interesting to apply *COPA* to other languages and domains where hardly any annotation for coreference resolution is available. In such cases, training on similar languages or relying more on the weak Internet features may all contribute.

List of Figures

List of Tables

List of Algorithms

Bibliography

Agarwal, Sameer, Jonwoo Lim, Lihi Zelnik-Manor, Pietro Perona, David Kriegman & Serge Belongie (2005). Beyond pairwise clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2, pp. 838–845.

Aone, Chinatsu & Scott W. Bennett (1995). Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, Mass., 26–30 June 1995, pp. 122–129.

Aronson, Alan R. (2001). Effective mapping of biomedical texts to the UMLS metathesaurus: The MetaMap program. In *In Proceedings of the AMIA Symposium 2001*, pp. 17–21.

Bagga, Amit & Breck Baldwin (1998). Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain, 28–30 May 1998, pp. 563–566.

Bansal, Mohit & Dan Klein (2012). Coreference semantics from web features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, 8–14 July 2012, pp. 389–398.

Bansal, Nikhil, Avrim Blum & Shuchi Chawla (2002). Correlational clustering. In *The proceeding of the 43rd annual symposium on foundations of computer science (FOCS)*, pp. 238–247.

Bar-Hillel, Aharon, Tomer Hertz, Noam Shental & Daphna Weinshall (2003). Learning distance functions using equivalence relations. In *Proceeding of 20th International Conference on Machine Learning*.

Basu, Sugato, Arindam Banerjee & Raymond J Mooney (2002). Semi-supervised clustering by seeding. In *Proceedings of International Conference on Machine Learning*, pp. 27–34.

Basu, Sugato, Mikhail Bilenko & Raymond J Mooney (2004). A probabilistic framework for semi-supervised clustering. In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 59–68.

Basu, Sugato, Ian Davidson & Kiri L. Wagstaff (Eds.) (2009). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Boca Raton, Flo.: CRC Press.

Ben-David, Shai, Ulrike von Luxburg & David Pal (2006). A sober look at clustering stability. In *Proceedings or the 19th Annual Conference on Learning Theory*, pp. 5–19. Berlin: Springer.

Bengtson, Eric & Dan Roth (2008). Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25-27 October 2008, pp. 294–303.

Blum, Avrim & Tom Mitchell (1998). Combining labeled and unlabeled data with Co-Training. In *Proceedings of the 11th Annual Conference on Learning Theory,* Madison, Wisc., 24–26 July, 1998, pp. 92–100.

Brennan, Susan E., Marilyn W. Friedman & Carl J. Pollard (1987). A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics,* Stanford, Cal., 6–9 July 1987, pp. 155–162.

Cai, Jie, Éva Mújdricza-Maydt, Yufang Hou & Michael Strube (2011a). Weakly supervised graph-based coreference resolution for clinical data. In *Proceedings of the 5th i2b2 Shared Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data, Washington,D.C.*

Cai, Jie, Éva Mújdricza-Maydt & Michael Strube (2011b). Unrestricted coreference resolution via global hypergraph partitioning. In *Proceedings of the 15th Conference on Computational Natural Language Learning,* Portland, Oreg., 23–24 June 2011.

Cai, Jie & Michael Strube (2010a). End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China, 23–27 August 2010, pp. 143–151.

Cai, Jie & Michael Strube (2010b). Evaluation metrics for end-to-end coreference resolution systems. In *Proceedings of the SIGdial 2010 Conference: The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue,* Tokyo, Japan, 24–25 September 2010, pp. 28–36.

Cardie, Claire & Kiri Wagstaff (1999). Noun phrase coreference as clustering. In *Proceedings of the 1999 SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora,* College Park, Md., 21–22 June 1999, pp. 82–89.

Charniak, Eugene & Mark Johnson (2005). Coarse-to-fine n-best parsing and MaxEent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Mich., 25–30 June 2005, pp. 173–180.

Chinchor, Nancy (2001). *Message Understanding Conference (MUC) 7*. LDC2001T02, Philadelphia, Penn: Linguistic Data Consortium.

Chinchor, Nancy & Beth Sundheim (2003). *Message Understanding Conference (MUC) 6*. LDC2003T13, Philadelphia, Penn: Linguistic Data Consortium.

Chomsky, Noam (1981). *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam (1995). *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Chung, Fan R.K. (1997). *Spectral Graph Theory*. Providence, R.I.: American Mathematical Society.

Coleman, Tom, James Saunderson & Anthony Wirth (2008). Spectral clustering with inconsistent advice. In *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 5–9 July 2008, pp. 152–159.

Culotta, Aron, Michael Wick & Andrew McCallum (2007). First-order probabilistic models for coreference resolution. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, N.Y., 22–27 April 2007, pp. 81–88.

Daumé III, Hal & Daniel Marcu (2005). A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., Canada, 6–8 October 2005, pp. 97–104.

Denis, Pascal & James Baldridge (2007). A ranking approach to pronoun resolution. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 6–12 January 2007, pp. 1588–1593.

Denis, Pascal & Jason Baldridge (2009). Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*, 42:87–96.

Dhillon, Inderjit S, Yuqiang Guan & Brain Kulis (2004). Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD)*, Vol. 2, pp. 551–556.

Fahrni, Angela, Vivi Nastase & Michael Strube (2012). HITS' cross-lingual entity linking system at TAC 2011: One model for all languages. In *Proceedings of the Text Analysis Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 14-15 November 2011.

Fellbaum, Christiane (Ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, Mass.: MIT Press.

Finkel, Jenny Rose, Trond Grenager & Christopher Manning (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Mich., 25–30 June 2005, pp. 363–370.

Finkel, Jenny Rose & Christopher Manning (2008). Enforcing transitivity in coreference resolution. In *Companion Volume to the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 15–20 June 2008, pp. 45–48.

Frank, Anette, Thomas Bögel, Oliver Hellwig & Nils Reiter (2012). Semantic annotation for the digital humanities – using Markov Logic Networks for annotation consistency control. *Linguistic Issues in Language Technology*, 7(8):1–21.

Grosz, Barbara J., Aravind K. Joshi & Scott Weinstein (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Grosz, Barbara J. & Candace L. Sidner (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Haghighi, Aria & Dan Klein (2007). Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 23–30 June 2007, pp. 848–855.

Haghighi, Aria & Dan Klein (2009). Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 6-7 August 2009, pp. 1152–1161.

Hahn, Udo & Michael Strube (1997). Centering in-the-large: Computing referential discourse segments. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and of the 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain, 7–12 July 1997, pp. 104–111.

Halkidi, Maria, Yannis Batistakis & Michalis Vazirgiannis (2001). On clustering validation techniques. *IEEE Intelligent Systems*, 17:107–145.

Hobbs, Jerry R. (1978). Resolving pronominal references. *Lingua*, 44:311–338.

Jain, Anil K., M.N. Murty & P.J. Flynn (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323.

Joshi, Aravind K. & Steve Kuhn (1979). Centered logic: The role of entity centered sentence representation in natural language inferencing. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, Tokyo, Japan, 20–23 August 1979, pp. 435–439.

Joshi, Aravind K. & Scott Weinstein (1981). Control of inference: Role of some aspects of discourse structure – centering. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, 24–28 August 1981, pp. 385–387.

Kamvar, Sepandar D., Dan Klein & Christopher D. Manning (2003). Spectral learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 9–15 August 2003, pp. 561–566.

Klein, Dan, Sepandar D. Kamvar & Christopher D. Manning (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceeding of the 19th International Conference on Machine Learning*.

Klenner, Manfred (2007). Enforcing consistency on coreference sets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria, 27–29 September 2007, pp. 323–328.

Kobdani, Hamidreza, Hinrich Schütze, Michael Schiehlen & Hans Kamp (2011). Bootstrapping coreference resolution using word associations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oreg., USA, 19–24 June 2011.

Kudoh, Taku & Yuji Matsumoto (2000). Use of Support Vector Machines for chunk identification. In *Proceedings of the 4th Conference on Computational Natural Language Learning*, Lisbon, Portugal, 13–14 September 2000, pp. 142–144.

Kulis, Brian, Sugato Basu, Inderjit Dhillon & Raymond Mooney (2005). Semi-supervised graph clustering: A kernel approach. In *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 7–11 August 2005, pp. 457–464.

Lang, Jun, Bing Qin, Ting Liu & Sheng Li (2009). Unsupervised coreference resolution with hpyergraph partitioning. *Computer and Information Science*, 2:55–63.

Lappin, Shalom & Herbert J. Leass (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu & Dan Jurafsky (2011). Stanfords multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the 15th Conference on Computational Natural Language Learning*, Portland, Oreg., 23–24 June 2011, pp. 28–34.

Luo, Xiaoqiang (2005). On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., Canada, 6–8 October 2005, pp. 25–32.

Luo, Xiaoqiang, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla & Salim Roukos (2004). A mention-synchronous coreference resolution algorithm based on the Bell Tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 21–26 July 2004, pp. 136–143.

MacQueen, J.B (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, pp. 281–297.

McCallum, Andrew & Ben Wellner (2005). Conditional models of identity uncertainty with application to noun coreference. In Lawrence K. Saul, Yair Weiss & Léon Bottou (Eds.), *Advances in Neural Information Processing Systems 17*, pp. 905–912. Cambridge, Mass.: MIT Press.

McCarthy, Joseph F. & Wendy G. Lehnert (1995). Using decision trees for coreference resolution. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montréal, Canada, 20–25 August 1995, pp. 1050–1055.

McCord, Michael C. (1989). Slot grammar: A system for simpler construction of practical natural language grammars. In *Natural Language and Logic'89*, pp. 118–145.

Milligan, Glenn W. & Martha W. Cooper (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.

Mitchell, Alexis, Stephanie Strassel, Shudong Huang & Ramez Zakhary (2004). *ACE 2004 Multilingual Training Corpus*. LDC2005T09, Philadelphia, Penn.: Linguistic Data Consortium.

Mitchell, Alexis, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstain, Lisa Ferro & Beth Sundheim (2002). *ACE-2 Version 1.0*. LDC2003T11, Philadelphia, Penn.: Linguistic Data Consortium.

Mitchell, Alexis, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstain, Lisa Ferro & Beth Sundheim (2003). *TIDES Extraction (ACE) 2003 Multilingual Training Data*. LDC2004T09, Philadelphia, Penn.: Linguistic Data Consortium.

Mitkov, Ruslan (2002). *Anaphora Resolution*. London, U.K.: Longman.

Müller, Christoph, Stefan Rapp & Michael Strube (2002). Applying Co-Training to reference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Penn., 7–12 July 2002, pp. 352–359.

Müller, Christoph & Michael Strube (2006). Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn & Joybrato Mukherjee (Eds.), *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pp. 197–214. Peter Lang: Frankfurt a.M., Germany.

Ng, Andrew Y., Michael J. Jordan & Yair Weiss (2002). On spectral clustering: Analysis and an algorithm. In T.G. Dietterich, S. Becker & Z. Ghahramani (Eds.), *Advances in Neural Processing Systems 14 (NIPS 2001)*, pp. 849–856. Cambridge, Mass.: MIT Press.

Ng, Vincent (2008). Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Waikiki, Honolulu, Hawaii, 25-27 October 2008, pp. 640–649.

Ng, Vincent (2010). Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 11–16 July 2010, pp. 1396–1411.

Ng, Vincent & Claire Cardie (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Penn., 7–12 July 2002, pp. 104–111.

Ng, Vincent & Claire Cardie (2003). Weakly supervised natural language learning without redundant views. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Alberta, Canada, 27 May –1 June 2003, pp. 173–180.

Nicolae, Cristina & Gabriel Nicolae (2006). BestCut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 22–23 July 2006, pp. 275–283.

Nigam, Kamal, Andrew Kachites McCallum, Sebastian Thrun & Tom Mitchell (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

NIST (2004). *The ACE evaluation plan: Evaluation of the recognition of ACE entities, ACE relations and ACE events*. http://www.itl.nist.gov/iad/mig//tests/ace/2004/doc/ ace04-evalplan-v7.pdf.

Pierce, David & Claire Cardie (2001). Limitations of Co-Training for natural language learning from large datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, Penn., 3–4 June 2001, pp. 1–9.

Ponzetto, Simone Paolo & Michael Strube (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, New York, N.Y., 4–9 June 2006, pp. 192–199.

Poon, Hoifung & Pedro Domingos (2008). Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Waikiki, Honolulu, Hawaii, 25-27 October 2008, pp. 650–659.

Pradhan, Sameer, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel & Nianwen Xue (2011). CoNLL-2011 Shared Task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Shared Task of 15th Conference on Computational Natural Language Learning*, Portland, Oreg., 23–24 June 2011.

Quinlan, J. Ross (1993). *C4.5: Programs for Machine Learning*. San Mateo, Cal.: Morgan Kaufman.

Raghavan, Preethi, Eric Fosler-Lussier & Albert M. Lai (2012). Exploring semi-supervised coreference resolution of medical concepts using semantic and temporal features. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 731–741.

Rahman, Altaf & Vincent Ng (2009). Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 6-7 August 2009, pp. 968–977.

Rahman, Altaf & Vincent Ng (2011). Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oreg., USA, 19–24 June 2011, pp. 814–824.

Rand, William R. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Recasens, Marta & Marta Vila (2010). On paraphrase and coreference. *Computational Linguistics*, 36(4):639–647.

Sapena, Emili (2012). *A constraint-based hypergraph partitioning approach to coreference resolution*, (Ph.D. thesis). Universitat Politècnica de Catalunya.

Sapena, Emili, Lluís Padró & Jordi Turmo (2010). A global relaxation labeling approach to coreference resolution. In *Proceedings of Coling 2010: Poster Volume*, Beijing, China, 23–27 August 2010, pp. 1086–1094.

Shi, Jianbo & Jitendra Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Song, Yang, Jing Jiang, Wayne Xin Zhao, Sujan Li & Houfeng Wang (2012). Joint learning for coreference resolution with Markov Logic. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, 8–14 July 2012.

Soon, Wee Meng, Hwee Tou Ng & Daniel Chung Yong Lim (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Stoer, Mechthild & Frank Wagner (1997). A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591.

Stoyanov, Veselin, Nathan Gilbert, Claire Cardie & Ellen Riloff (2009). Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, Singapore, 2–7 August 2009, pp. 656–664.

Strube, Michael (1998). Never look back: An alternative to centering. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, Montréal, Québec, Canada, 10–14 August 1998, Vol. 2, pp. 1251–1257.

Strube, Michael & Udo Hahn (1999). Functional centering: Grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.

Tibshirani, Robert, Guenther Walther & Trevor Hastie (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Toutanova, Kristina, Dan Klein, Christopher D. Manning & Yoram Singer (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Alberta, Canada, 27 May –1 June 2003, pp. 252–259.

Uzuner, Özlem, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian & Brett R South (2012). Evaluating the state of the art in coreference resolution for electronic medical records. In *J Am Med Inform Assoc. Published online first: 24 February 2012 doi:10.1136/amiajnl-2011-000784*.

Versley, Yannick, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang & Alessandro Moschitti (2008). BART: A modular toolkit for coreference resolution. In *Companion Volume to the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 15–20 June 2008, pp. 9–12.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly & Lynette Hirschman (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pp. 45–52. San Mateo, Cal.: Morgan Kaufmann.

von Luxburg, Ulrike (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

von Luxburg, Ulrike (2010). Clustering stability: An overview. *Foundations and Trends in Machine Learning*, 2(3):235–274.

Wagstaff, Kiri (2002). *Intelligent clustering with instance-level constraints*, (Ph.D. thesis). Cornell University. Ch5: constrained clustering application to coreference resolution.

Wagstaff, Kiri & Claire Cardie (2000). Clustering with instance-level constraints. In *Proceedings of the 17th International Conference on Machine Learning*, Palo Alto, Cal., 29 June – 2 July 2000, pp. 1103–1110.

Walker, Marilyn A. (1998). Centering, anaphora resolution, and discourse structure. In M.A. Walker, A.K. Joshi & E.F. Prince (Eds.), *Centering Theory in Discourse*, pp. 401–435. Oxford, U.K.: Oxford University Press.

Weischedel, Ralph, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin & Ann Houston (2011). *OntoNotes Release 4.0*. LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium.

Witten, Ian H. & Eibe Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco, Cal.: Morgan Kaufmann.

Wu, Zhenyu & R Leahy (1993). An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113.

Xing, Eric P, Andrew Y Ng, Michael I Jordan & Stuart Russell (2003). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems(NIPS 2003)*.

Yang, Xiaofeng, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu & Sheng Li (2008). An entity-mention model for coreference resolution with Inductive Logic Programming. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio, 15–20 June 2008, pp. 843–851.

Yang, Xiaofeng, Jian Su & Chew Lim Tan (2005). A twin-candidate model of coreference resolution with non-anaphor identification capability. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, Jeju Island, South Korea, 11–13 October 2005, pp. 719–730.

Yu, Stella X & Jianbo Shi (2004). Segmentation given partial grouping constraints. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. vol. 26, No. 2.

Zhou, Dengyong, Jiayuan Huang & Bernhard Schölkopf (2007). Learning with hypergraphs: Clustering, classification, and embedding. In B. Schölkopf, J. Platt & T. Hofmann (Eds.), *Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, pp. 1601–1608. Cambridge, Mass.: MIT Press.

Zien, Jason Y., Martine Schlag & Pak K. Chan (1999). Multi-level spectral hypergraph partitioning with arbitrary vertex sizes. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, pp. 1389–1399.